Introduction to JavaScript

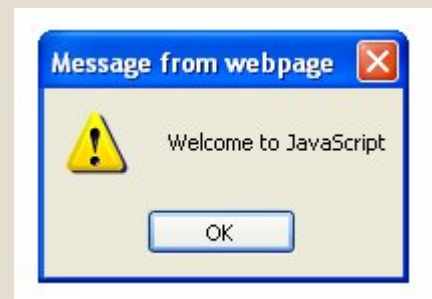
جاوا اسکرپٹ ایک Web Programing Language ہے یہ سن
1995 میں وجود میں آئی ہے جاوا اسکرپٹ Client Side
Programing Language ہے یہ HTML میں Head Tag کے اندر
لکھی جا تی ہے آپ چاہیں تو اس کو Body Tag کے اندر بھی لکھ
سکتے ہیں لیکن صحیح طریقہ یہ ہے کے آپ اس کو agHead T کے
اندر ہی لکھیں - جاوا اسکرپٹ کا کوڈ Browser پڑھتا یعنی
Interprete کرتا ہے یہ بہت ہلکی یعنی Lightweight Language
ہے ہر Browser اس کو Support کرتا ہے اس لیے یہ Cross
Browser کہلاتی ہے جاوا اسکرپٹ بہت Powerful Language ہے
اور یہ HTML Tag کو بھی پڑھتی ہے جاوا اسکرپٹ کو لکھنے کے
لیے آپ کو Head Tag کے اندر Script Tag لگانا ہو گا - جاوا
اسکرپٹ Case Sensitive Language ہے آگے جا کر آپ کو اس
کی مثال بھی دیں گے - جب آپ JavaScript کو Head Tag کے اندر
لکھتے ہیں تو یہ Internal JavaScript کہلائے گی اور اگر آپ ایک
الگ File میں لکھیں تو یہ External JavaScript کہلائے گی اس
کی مثال آپ کو آگے جا کر تفصیل سے دیں گے اور External File
کا js.Extension ہوتا ہے - JavaScript کو ہم نے بہت محنت سے

بیان کرنے کی کوشش کی ہے پھر بھی اگر کوئی کمی محسوس ہو تو آپ ہم کو Email کرسکتے ہیں -

Next

Write your first JavaScript

اب ہم آپ کو JavaScript کا پہلا کوڈ لکھنا سکھاتے ہیں جس میں صرف MessageBox Show ہو گا آپ Head Tag کے اندر نیچے دیا ہوا کوڈ لکھیں JavaScript کو لکھنے لے لیے آپ کو Script کی Type بتانی ہو گی اور پھر Script کا Tagبند کرنا ہوگا -

```
<script type="text/javascript">
   alert('Welcome to JavaScript');
</script>
```

Alert Function

Alert فنکشن یوزر کو Message دکھانے کے لیے استعمال کیا جاتا ہے بریکٹ میں جو الفاظ Words لکھیں ہوئے ہیں وہ Function کا

Parameter کہلاتا ہے اس کے اندر آپ جو کچھ بھی لکھیں گے وہ ویسا ہی MessageBox میں ظاہر ہو گا -

Next | Previous

اگر Parameter میں Number Value لکھنی ہو تو آپ اس کو بغیر Single Quote یا eDouble Quot کے لکھیں گے جیسا کے نیچے دی گئی مثال میں آپ دیکھ سکتے ہیں کہ دو نمبروں کو جمع کیا گیا ہے -

```
<script type="text/javascript">
      alert(5+4);
</script>
```

Alert Function ایک ایسا Function ہے جو نہ صرف Text کو Print کرتا ہے بلکہ آپ اس سے ulationCalc بھی کر سکتے ہیں یاد رہے کہ JavaScript میں ہر لاین کے آخر میں Semi Colon لگاتے ہیں اس کو Line Termination بھی کہا جاتا ہے -

document.write()

یہ Browser Function پر Text Print کرتا ہے یعنی آپ Browser پر با آسانی لکھ سکتے ہیں اب اس کی مثال دیکھتے ہیں -

Welcome to JavaScript.

Next | Previous

```
<script type="text/javascript">
       document.write('Welcome to JavaScript');
```

```
</script>
```

اس کے علاوہ آپ اس Function کی مدد سے Calculation بھی کر سکتے ہیں -

```
<script type="text/javascript">
    document.write(7+2);
</script>
```

Comments in JavaScript

اگر آپ یہ چاہتے ہیں کہ آپ کا کوڈ لکھا بھی رہے اور Execute بھی ناہو اس کے لیے آپ کو کوڈ کو Comments کرنا ہو گا -
Comments دو طرح کے ہوتے ہیں ایک Single Line Comments اور دوسرا Multiple Line Comments اب ان دونوں کی مثال دیکھتے ہیں -

Next | Previous

Single Line Comments

اگر آپ ایک Line کو Comment کرنا چاہتے ہیں اس کے لیے آپ کو Single Line Comment کو استعمال کرنا ہوگا دو دفعہ Back Slash لگانے سے آپ ایک لاین کو Comment کر سکتے ہیں اب آپ پیج کو Save کریں اور Browser پر اس کا Result دیکھیں گے تو آپ کو Comments والا Text نظر نہیں آئے گا -

```
<script type="text/javascript">
```

```
        //document.write(7+2);
    </script>
```

Multiple Line Comments

اگر ایک سے ذیادہ لاینوں کو Comments کرنا چاہتے ہیں اس کے لیے آپ کو Multiple Line Comments کو استعمال کرنا ہو گا جیسا کہ نیچے دکھایا گیا ہے -

```
    <script type="text/javascript">
        /*document.write(7+2);
          alert('I am in Comments');*/
    </script>
```

Next | Previous

اس کو لگانے کا طریقہ یہ ہے کہ آپ ایک دفعہ Back Slash اور Asterik لگایں اور جہاں تک آپ کو Comments کرنا ہے وہاں لاین کے آخر میں جا کر Asterik اور Back Slash لگا دیں آپ کا کوڈ Comments ہو جائے گا پچھلی دی گئی مثال میں با آسانی دیکھ سکتے ہیں اب آپ اپنے Browser پر اس کا Result دیکھیں گے تو آپ کو کچھ نظر نہیں آئے گا اور جب آپ کو Code کی ضرورت ہو تو آپ اپنے کوڈ کو Comments سے ہٹا سکتے ہیں اور استعمال کر سکتے ہیں -

Semi Colon

JavaScript میں Semi Colon لگانا Optional ہے لیکن لگانا صحیح طریقہ ہے اور یہ Case Sensitive Language کی پہچان بھی ہے کہ ہر لاین کے آخر میں Semi Colon لازمی آتا ہے -

Case Sensitive

Alphabate دو طرح کے ہوتے ہیں ایک Capital Letter اور دوسرا Small Letter جتنی بھی Case Sensitive Language ہیں ان میں دونوں بڑا A اور چھوٹا a کو الگ الگ سمجھتے ہیں کیونکہ دونوں کے ASCII Code الگ ہیں -

Next | Previous

HTML in JavaScript

JavaScript آپ کے HTML Tag کو بھی پڑھتی ہے نیچے دی گئی مثال میں آپ دیکھ سکتے ہیں -

```
<script type="text/javascript">
     document.write('<h1>This is heading1.</h1>');
</script>
```

This is heading1.

Variable in JavaScript

ڈیٹا کو مختصر عرصہ تک محفوظ کرنے کے لیے Variable استعمال کرتے ہیں یعنی جب تک آپ کا Browser Open ہے تب تک

Variable میں ڈیٹا ہے جیسے ہی آپ نے اپنا Browser Close کیا تو تمام ڈیٹا اور Variable ختم ہو جایئں گے Variable کو بنانے کے لیے آپ کو سب سے پہلے var لکھنا ہو گا اس کے بعد Variable کا نام لکھتے ہیں اور برابر کی علامت لگاتے ہیں پھر Value دیتے ہیں اور آخر میں Semi Colon لگاتے ہیں -

Next | Previous

نیچے دی گئی مثال میں ایک Variable بنایا گیا ہے جس کا نام myname ہے اور اس کی Value Ahmed ہے اس کا مطلب یہ ہے کہ myname کے اندر Ahmed محفوظ کرنا یا ڈالنا ہے - Variable بناتے وقت var لکھنا Optional ہے اگر نہیں لکھیں گے تو جب بھی Variable بن جائے گا -

```
<script type="text/javascript">
     var myname = 'Ahmed';
</script>
```

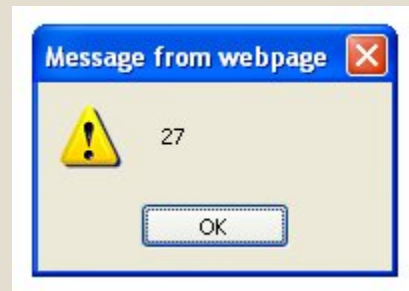
Data Type in JavaScript

JavaScript ایک Loose Data Type Language ہے یعنی اس میں بتانا نہیں پڑتا ہے کہ اس Variable میں کس قسم کا ڈیٹا آئے گا پچھلی مثال میں آپ دیکھ چکے ہیں کے ایک Variable میں شخص کا نام Store کیا تھا اور اس نام کو Single Quotation کے اندر لکھا تھا یاد رہے کہ جو Value Inverted Commas کے اندر لکھی جاتی

ہے وہ String Value کہلاتی ہے پچھلے کوڈ کی String Value تھی اگلے پیج پر نمبر Value کی مثال کو دیکھیں گے -

Next | Previous

JavaScript میں ڈیٹا کی Type بتانے کی ضرورت نہیں ہوتی ہے یہ Value کو خود بخود Convert کر دیتا ہے - JavaScript میں عموما تین قسم کی Data Type ذیادہ استعمال ہوتی ہیں جن میں String, Number اور Data TypeBoolean شامل ہیں - نیچے دی گئی مثال Number Data Type کی ہے -

```
<script type="text/javascript">
    var age = 27;
    alert(age);
</script>
```

آپ نے دیکھا کہ Variable میں Value Store کرنے کے بعد Alert Function میں اس کو Show کروایا گیا ہے اور صرف Variable

کا نام دیا ہے جو بھی Variable Value میں ہو گی وہ
MessageBox میں Show ہو جائے گی -

Next | Previous

Variable Rules

Variable کو بنانے کے کچھ اصول ہوتے ہیں جن پر عمل کرنا
لازمی ہے سب سے پہلے ایک بات یاد رکھیں کے Variable میں
Blank Space نہی آتا ہے Variable کا پہلا حروف نمبر سے شروع
نہیں ہوتا ہے - myname اور myName دونوں الگ الگ Variable
ہیں - Variable کا نام کبھی بھی Keyword یا Reserve Word
نہیں ہو گا Variable کے درمیان اگر Space دینا ہو تو لازمی
Underscore استعمال کرنا ہوگا - Variable کا پہلا حرف Small
Letter سے شروع ہو گا نام کے درمیان Underscore اس طرح سے
لگانا ہے جیسے نیچے دکھایا گیا ہے -

```
var my_name = 'Abdullah';
```

Variable کی خاص بات یہ ہے کہ Variable ایک وقت میں ایک
Value Store کرتے ہیں - نیچے دی گئی مثال کو دیکھیں -

```
var age = 27;
age = 10;
age = 45;
age = 18;
alert(age);
```

Next | Previous

آپ دیکھ رہے ہیں کہ ایک Variable میں ایک سے ذیادہ Value Store کی گئی ہیں لیکن MessageBox میں آخری Value آئے گی کیونکہ ہر لاین پر Value تبدیل ہو رہی ہے جو آخری Value ہو گی وہی Current Value کہلائے گی ایک لاین میں ایک سے ذیادہ Variable بھی بنائے جا سکتے ہیں جیسا نیچے دکھایا گیا ہے -

```
    var fnam, lname, age, email, location;
    var age=18, email='yourname@gmail.com',
location='Microsoft';
```

Variable Declaration

Variable کو بغیر Value کے بنانا Declaration کہلاتا ہے آپ چاہیں تو بعد میں Value دے سکتے ہیں نیچے دی گئی مثال میں Value نہیں دی گئی ہے صرف Variable کو Declare کیا گیا ہے -

```
   var my_name;
```

Next | Previous

Variable Assignment

جب آپ کسی Variable کو Value دیتے ہیں تو یہ عمل Assignment کہلاتا ہے پچھلی مثال میں صرف Declare Variable ہوا ہے اب اس مثال میں اس کی Value Assign کی جا رہی ہے -

```
    var my_name = 'Abdullah';
```

یا آپ اس کو اس طرح بھی لکھ سکتے ہیں پہلے Declare اور پھر Assign کریں -

```
var my_name;
my_name = 'Abdullah';
```

String Number

آپ پڑھ چکے ہیں کے String Value کو ہمیشہ Single Quotation یا Double Quotation میں لکھا جاتا ہے لیکن اگر Number String میں ہوں تو آپ اس کو جمع کیسے کریں گے -

```
10 + 5 =15
10 + "5" = 105
8 + 8 + "8" = 168
```

Next | Previous

پچھلی مثال میں آپ دیکھ رہے ہیں کہ جب دس کے ساتھ پانچ کو ملایا گیا تو وہ جمع ہونے کے بجائے مل کر جمع ہو گئے ہیں اور جواب 105 آیا اور جب تین ویلیو کو ملایا گیا تو دو ویلیو جمع ہو گئی اور باقی ایک مل گئی ہے اور جواب 168 آگیا اب آپ سمجھ گئے ہونگے کہ اگر نمبر String میں ہو تو نتیجہ کیا آئے گا -

String Concatenation

اگر آپ دو یا دو سے ذیادہ Variable کو ملانا چاہیں تو اس کے لیے آپ کو Plus Sign استعمال کرنا ہو گا اس کو String Concatenation کہتے ہیں -

```
var first_name = 'Mohammad';
var last_name = 'Ahmed';
alert(first_name + last_name);
```

جب آپ اوپر دیئے گئے کوڈ کو چلایں گے تو دونوں Value جمع ہو کر MessageBox میں Show ہو جایئں گی اب ایک مسلہ ہو گا کے درمیان میں گیپ نہیں ہو گا گیپ دینے کے لیئے Inverted Commas میں Space Bar سے خالی اسپیس دیں گے -

```
alert(first_name + " " + last_name);
```

Next | Previous

اب آپ Number Value کے ساتھ Concatenate کر کے براوزر پر دیکھیں -

```
var age = 18;
alert('Your age is ' + age);
```

If Statement

جیسا کے نام سے ظاہر ہے کہ If کا مطلب اگر کے ہیں If Statement کو Condition کے ساتھ استعمال کیا جاتا ہے اس کے ساتھ مختلف قسم کے Operator بھی استعمال ہوتے ہیں - If

Statement کا ایک بلاک ہوتا ہے اس بلاک میں آپ جو کچھ لکھیں گے وہ کوڈ If Statement کی Condition کے True ہونے پر Execute ہو گا اور بلاک بنانے کے لیئے Curly Brackte استعمال کیے جاتے ہیں -

Next | Previous

Comparision Operator

دو دفعہ == برابر کی علامت لگانے کو Comparision Operator کہا جاتا ہے اس کو If Statement کے ساتھ لکھا جاتا ہے اب اس کو لکھنا سیکھتے ہیں -

```
<script type="text/javascript">
  var my_name = 'Ameen';
  if(my_name == 'Ameen')
  {
     alert('Hi Mr. Ameen');
  }
</script>
```

اگر If Statement کی Condition True ہو گی تو If بلاک کا کوڈ چلے گا اور اگر False ہوگی پھر کچھ نہیں ہو گا If کے بعد Braket لگاتے ہیں اور پھر Braket کے اندر Condition لگائی جاتی ہے اس کے بعد If کا بلاک بنایا جاتا ہے جیسا کہ اوپر دی گئی مثال میں دکھایا گیا ہے -

Next | Previous

```
<script type="text/javascript">
```

```
      var my_name = 'Ameen';
      if(my_name == 'Ameen1')
      {
        alert('Hi Mr. Ameen');
      }
   </script>
```

اوپر دی ہوی Condition False ہے اس لیے کوڈ نہیں چلے گا کیونکہ Variable کی Value Ameen ہے اور اس کو 1Ameen سے Match یا Compare کرایا گیا ہے لہذہ یہ Condition False ہے -

Else Statement

پچھلی مثال میں آپ نے دیکھا کہ اگر Condition False تھی تو If کا کوڈ Execute نہیں ہوا تھا لیکن اگر آپ یہ چاھتے ہیں کہ Condition False ہونے کی صورت میں بھی کوڈ Run ہو تو اس کے لیئے آپ کو Else Statemen استعمال کرنا ہو گا یہ If Statement کے بعد لکھا جاتا ہے اگلے پیج پر Else Statement کی مثال دیکھیں -

Next | Previous

```
<script type="text/javascript">
      var my_name = 'Ameen';
      if(my_name == 'Ameen1')
      {
         alert('Hi Mr. Ameen');
      }
      else
```

```
    {
       alert('I dont know');
    }
  </script>
```

اوپر دی ہوئی مثال میں If Condition False ہے اس لیئے Else Block کا کوڈ Execute ہوگا اور MessageBox میں I dont know آئے گا -

Escape Sequance

JavaScript میں Escape Sequence موجود ہیں جو String Value میں استعمال ہوتے ہیں جیسے اگر آپ Single Quote کو استعمال کرنا چاہیں اس کے لیئے آپ کو Slash کے ساتھ Single Quote لگانا ہو گا اگلے پیج پر Escape Sequance کی Examples دی گئی ہیں -

Next | Previous

Single Quote

اسٹرنگ ویلیو کے اندر Single Quote لگانے کے لیئے پہلے Double Quote میں اسٹرنگ کو لکھیں پھر اس کے اندر جس الفاظ کو Single Quote میں لکھنا ہے اس کو سلیش اور Single Quote میں Open کریں اس کے بعد پھر اس کو Single Quote اور سلیش میں Close کریں جیسا کہ نیچے دکھایا گیا ہے -

```
document.write("Welcome to \'JavaScript\'.");
```

Welcome to 'JavaScript'.

Double Quote

اس کے لیئے بھی وہی طریقہ ہے جو Single Quote کا ہے صرف Single Quote کی جگہ آپ نے Double Quote استعمال کرنا ہے نیچے دی گئی مثال کو دیکھیں -

```
document.write('Welcome to \"JavaScript\".');
```

Welcome to "JavaScript".

Next | Previous

Back Slash

اگر String Value میں Back Slash لگانا چاہیں تو Double Slash لگا نا ہو گا جیسے نیچے دی گئی مثال میں دکھایا گیا ہے -

```
document.write("Welcome \\to JavaScript.");
```

Welcome \to JavaScript.

Operator Precedence

اگر Calculation میں + - x / ہو رہا ہو تو سب سے پہلے کونسی علامت حل ہو گی یاد رہے کہ سب سے پہلے Divide ہو گا اس کے بعد Multiply اور پھر Add اور آخر میں Subtract حل ہو گا پہلے 9 / 3 ہو گا پھر جو جواب آئے گا وہ 6 سے ضرب ہو گا اس کے بعد 5

2 + کا جواب 7 آئے گا اور جو جواب ضرب ہونے کے بعد آیا تھا اس میں سے 7 کو نکال دیں آخر میں جواب 5 آئے گا -

```
document.write(2 + 5 - 6 * 3 / 9);
```

Next | Previous

لیکن اگر آپ یہ چاہتے ہیں کے پہلے جمع کی علامت حل ہو اس کے بعد دوسری علامات حل ہوں اس کے لیئے آپ کو () Bracket کو استعمال کرنا ہو گا نیچے دی گئی مثالوں کو سمجھنے کو کوشش کریں ہر مثال میں سب سے پہلے Bracket کے اندر والی Values حل ہونگی اس کے بعد دوسری Values ترتیب سے حل ہوتی جایئں گی -

```
document.write((2 + 5) - 6 * 3 / 9);
document.write(2 + (5 - 6) * 3 / 9);
document.write(2 + 5 - (6 * 3) / 9);
```

The + Operator

JavaScript میں Plus Sign کو آپ Number Values کو جمع کرنے کے لیئے بھی استعمال کر سکتے ہیں اور String Values کو ملانے کے لیئے بھی استعمال کرتے ہیں اگر String Values میں ہونگی تو Plus Sign ویلیو کو Concatenate کر دیگا اور اگر Number Values میں ہونگی تو جمع کر دیگا مثال نیچے دی گئی ہے -

```
var ans = 2 + 5 + 1;
```

```
    var name = 'Mohammad' + 'Ali';
```

Next | Previous

Arithmetic Expression

حسابی مساوات کو Arithmetic Expression کہتے ہیں اس میں + -
x / کرتے ہیں -

```
  var ans = 2 + 5 + 1;
```

Comparision Operator

یہ وہ Operator ہوتے ہیں جو Condition کو چیک کرتے ہیں کہ
Condition True ہے یا False جیسے 1 < 2 اس کا مطلب یہ ہے کہ
ایک چھوٹا ہے دو سے یہ True Condition ہے -

Description	Operator
Left side is equal to right side	==
Left side is less than right side	<
Left side is greater than right side	>
Left side is less than or equal to right side	<=
Left side is greater than or equal to right side	>=
Left side is not equal to right side	!=

Next | Previous

پچھلے پیج پر تمام Operator کی مثالیں آگے بیان کی گئی ہیں اب ان
تمام کو ایک ایک کر کے دیکھتے ہیں اور سمجھنے کی کو شش
کرتے ہیں -

Less Than Operator

یہ Operator ایک ueVal کو دوسری Value سے کم کی طرف ظاہر کرتا ہے جیسے2 >1 یعنی 1 چھوٹا ہے 2 سے یہ Condition True ہے - نیچے دی ہوئی مثال کو Execute کرکے دیکھیں -

```
var age = 24;
if(age < 25)
{
    alert('This is true condition');
}
else
{
    alert('This is false condition');
}
```

Next | Previous

Greater Than Operator

یہ Operator ایک Value کو دوسری Value سے بڑی یا ذیادہ کی طرف ظاہر کرتا ہے اس Operator میں بڑی Value کو چیک کیا جاتا ہے جیسے 2 < 1 اس کو اس طرح پڑھتے ہیں 1 بڑا ہے 2 سے یہ Condition False ہے نیچے دی گئی مثال کو دیکھیں اس میں Condition True ہے یا eFals ہے - اگر Condition True ہے تو If Block کا کوڈ Execute ہوگا ورنہ Else Block کا کوڈ Execute ہوگا -

```
var age = 24;
if(age > 25)
```

```
    {
       alert('This is true condition');
    }
    else
    {
       alert('This is false condition');
    }
```

Next | Previous

Less than or equal to operator

یہ Operator دو Condition Check کرتا ہے ایک ویلیو کم ہو یا دوسری برابر ہو اس سے آپ کم یا برابر Value کو چیک کرتے ہیں -

```
    var age = 25;
    if(age <= 25)
    {
        alert('This is true condition');
    }
    else
    {
        alert('This is false condition');
    }
```

اوپر دی ہوئی مثال میں آپ ویلیو کو تبدیل کر کے بھی چیک کریں تاکہ آپ کو اچھی طرح سے سمجھ میں آجائے -

Next | Previous

Greater than or equal to operator

یہ Operator بھی دو Condition Check کرتا ہے کہ Value بڑی ہو یا برابر ہو اس سے آپ زیادہ یا برابر Value کو چیک کرتے ہیں -

```
var age = 26;
if(age >= 25)
{
    alert('This is true condition');
}
else
{
    alert('This is false condition');
}
```

اوپر دی گئی مثال میں ویلیو کو بھی تبدیل یعنی کم یا ذیادہ کر کے اس کا Browser Output پر دیکھیں -

Next | Previous

Not Equal to Operator

جیسا کے نام سے ظاہر ہے کہ ایسی Condition جو دی ہوئی Value کے برابر نہیں ہے یعنی کچھ بھی Value ہو لیکن اس کے برابر نہیں ہو - اگر True Condition ہے تو If Block کا کوڈ Execute ہوگا ورنہ Else Block کا کوڈ Execute ہوگا -

```
var age = 26;
if(age != 25)
{
    alert('This is true condition');
}
```

```
    else
    {
        alert('This is false condition');
    }
```

Next | Previous

Logical Operator

JavaScript میں Logical Operator تین قسم کے ہوتے ہیں اور یہ بھی Condition پر کام کرتے ہیں اب ان تینوں کو Detail میں مثال کے ساتھ سمجھتے ہیں -

i. And Operator
ii. Or Operator
iii. Not Operator

And Operator

یہ Operator دو Condition کو ایک ہی وقت میں چیک کرتا ہے یعنی جب تک دونوں Condition True نہ ہوں جواب True نہیں ہو گا دونوں میں سے کوئی ایک Condition False ہوئی تو جواب False ہو گا اس کو اگلے پیج پر Table کی مدد سے سمجھتے ہیں -

Next | Previous

نیچے دیئے گئے Table میں صرف ایک Condition True ہے باقی سب Condition False ہے کیونکہ And Operator کی یہ شرط ہے

کہ دونوں Condition کا True ہونا ضروری ہے اب اس کی مثال دیکھتے ہیں -

Result	Condition 2	Condition 1
True	True	True
False	False	False
False	False	True
False	True	False

```
var username = 'admin';
var password = 12345;

if(username == 'admin' && password == 12345)
{
   alert('Login Successfully.');
}
else
{
   alert('Invalid username or password.');
}
```

Next | Previous

پچھلی مثال میں دو Values کو ایک ساتھ چیک کیا گیا ہے پہلی username اور دوسری password ہے اور یہ دونوں Condition True ہیں یاد رہے کہ And Operator کو دو دفعہ && ایمپرسینڈ Ampersend لگا کر لکھا جاتا ہے -

Or Operator

یہ اوپریٹر And Operator کے مخالف ہے اس میں کوئی بھی ایک True Condition ہو گی تو جواب True ہو گا اور اگر دونوں False ہو نگی تو جواب False ہو گا -

```
var username = 'admin';
var password = 12345;

if(username == 'admin' || password == 123456)
{
   alert('Login Successfully.');
}
else
{
   alert('Invalid username or password.');
}
```

Next | Previous

پچھلی مثال میں پہلی Condition True ہے اور دوسری Condition False ہے کیونکہ password کی Value Match نہیں ہو رہی ہے لیکن username کی Value Match ہو رہی ہے اس لیئے پہلی Condition True ہے لہذا MessageBox میں جواب Login Successfully آئے گا -

نیچے Or Operator کا ٹیبل دیا ہوا ہے جس میں صرف ایک True Condition ہے اور باقی سب False Conditions ہیں یاد رہے کے

Or Operator کو || کی طرح لکھا جاتا ہے اس کو Pipe Sign بھی کہتے ہیں -

Result	Condition 2	Condition 1
True	True	True
False	False	False
True	False	True
True	True	False

Next | Previous

Not Operator

جیسا کہ نام سے ظاہر ہے کے Not کا مطلب انکار کے ہیں یعنی ایک ایسی Condition جس میں اس Value کے سوا کچھ بھی ہو یا دوسرے الفاظ میں یہ Condition نا ہو یعنی اگر کوئی شخص یہ کہے کے اگر مجھے گرمی نہیں لگے گی تو پھر میں Soup پیونگا - If Condition میں جو Condition دی ہے یہ اس کے Opposite یعنی مخالف Condition کو چیک کرے گا - Not Operator کے لیئے اس ! Sign کو استعمال کرتے ہیں - نیچے دی ہوئی مثال میں Variable Value 10 سے بڑی ہے تو If کی Condition False ہوگی لیکن Not Operator نے اس کو True Condition بنا دیا ہے -

```
var weatherDegree = 15;
if(!(weatherDegree < 10))
{
  alert('Normal Weather');
```

```
    }
    else
    {
        alert('Winter Season');
    }
```

Next | Previous

نیچے دیا ہوا ٹیبل Not Operator کا ہے اگر If Condition True ہوگی تو وہ False ہوجائے گی اور اگر If Condition False ہوگی تو True ہو جائے گی -

Result	Condition
False	True
True	False

Loops in JavaScript

Programing میں ایک کام کو بار بار کرنے کے لیئے Loop Statement کو استعمال کرتے ہیں مثلا ایک نام کو 100 دفعہ لکھنا یا 1 سے 100 تک گنتی لکھنا وغیرہ - Loop Statement تین طرح کے ہوتے ہیں For Loop دوسرا While Loop اور تیسرا Do While Loop ہوتا ہے اب اس کی مثالیں دیکھتے ہیں -

Next | Previous

For Loop in JavaScript

اس Loop کے تین Parameter ہوتے ہیں ایک Starting Value دوسرا Condition اور تیسرا Ending Value ہوتا ہے - سب سے

پہلے ایک Variable لیا گیا ہے جس کو Loop میں استعمال کریں گے اس کے بعد اس کی Value Zero دی گئی ہے یہ پہلا Parameter ہے اور ہر Parameter کے بعد Semi Colon آتا ہے اور دوسرے Parameter میں Condition آتی ہے یعنی Loop کو کس Value تک چلانا ہے یا پھر Loop کو کس Value پر لا کر ختم کرنا ہے آخری تیسرے Parameter میں اضافہ Increament آتا ہے جب پہلی دفعہ Loop چلے گا تو اس کی Value Zero ہو گی اس کے بعد Condition چیک ہو گی اگر Condition صحیح ہے تو Variable کی Value میں ایک کا اضافہ ہو گا اور Curly Bracket کے اندر کا کوڈ چلے گا اس طرح سے جب تک Condition True ہے ہر دفعہ 1 کا اضافہ ہوتا چلا جائے گا اور Variable کی Value Zero سے دی ہوئی Condition تک ہو جائے گی اگلے پیج پر For Loop کی مثال دیکھیں اور اس کو سمجھنے دی کوشش کریں -

Next | Previous

جب آپ نیچے دیئے ہوئے Loop کو چلایں گے تو گیارہ دفعہ Javascript کو پرنٹ کرے گا کیونکہ یہ Zero سے شروع ہوا ہے اور Condition میں دو Condition دی ہوئی ہے یعنی Value 10 دس سے چھوٹی ہو یا برابر ہو اس طرح سے Variable کی ویلیو 0 سے 10 تک ہوگی اور Loop گیارہ دفعہ JavaScript کو پرنٹ کرے

گا - یاد رہے کہ 2 دفعہ ++ کا مطلب Variable Value میں 1 کا اضافہ کرنا ہے -

```
var i;
for(i=0; i<=10; i++)
{
   document.write('Javascript');
}
```

دوسری مثال کو دیکھیں اس کی Condition میں Variable کی Value دس سے چھوٹی رکھی ہے جب کہ اوپر کی مثال میں Condition دس سے چھوٹی یا برابر ہے اس لیئے یہ Loop 0 سے 9 تک چلے گا اور 10 دفعہ JavaScript لکھے گا -

```
var i;
for(i=0; i < 10; i++)
{  document.write('Javascript');   }
```

Next | Previous

اسی Loop سے آپ گنتی لکھنا سیکھیں گے نیچے دی گئی مثال میں 1 سے 10 تک گنتی لکھی گئی ہے - جب پہلی دفعہ Loop چلے گا تو Variable کی Value 1 ہوگی اور 1 پرنٹ ہو جائے گا دوسری دفعہ چلے گا تو Value 2 ہوگی کیونکہ ہر دفعہ 1 کا اضافہ کیا گیا ہے اس طرح سے یہ Loop 10 تک گنتی لکھے گا -

```
var i;
for(i=1; i<=10; i++)
{
   document.write(i);
```

```
}
```

اگر آپ گنتی کو Vertical لکھنا چاہتے ہیں تو آپ کو Variable کے ساتھ Inverted Commas کے اندر Br کے Tag کو Concatenate کرنا ہو گا اور ہر دفعہ Value Print ہونے کے بعد اگلی ویلیو Next Line سے Start ہوگی -

```
var i;
for(i=0; i<=10; i++)
{
   document.write(i + '<br />');
}
```

Next | Previous

Loop کو جہاں سے چاہیں شروع کر سکتے ہیں نیچے دی گئی مثال میں آپ Loop کو 3 سے شروع اور 9 پر ختم کریں گے اس طرح سے یہ 7Loop دفعہ چلے گا -

```
var i;
for(i=3; i <= 9; i++)
{
   document.write(i + '<br/>');
}
```

Reverse Loop

For Loop کو آپ Reverse بھی لکھ سکتے ہیں یعنی 10 سے 1 تک Reverse یا الٹی گنتی لکھ سکتے ہیں - ہر Parameter کو

Opposite لکھنا ہو گا نیچے دی گئی مثال کو سمجھنے کی کوشش کریں -

```
var i;
for(i = 10; i >= 1; i--)
{
   document.write(i + '<br/>');
}
```

Next | Previous

Math Table in JavaScript

اب آپ For Loop کو استعمال کرتے ہوئے 2 کا پہاڑا لکھنا سیکھیں گے پچھلی مثال میں 1 سے 10 تک گنتی پرنٹ ہو رہی تھی اس مثال میں جو Value Fix ہیں ان کو Variable کے ساتھ Single Quote میں لکھنا ہے جیسا کہ نیچے دی گئی مثال میں دکھایا گیا ہے -

```
var i;
for(i=1; i <= 10; i++)
{
   document.write('2 x ' + i + ' = ' + 2 * i +
'<br/>');
}
```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

جو Value Fix ہے اس کو Single Quotation میں لکھا گیا ہے تاکہ یہ گنتی کے ساتھ As it is پرنٹ ہوجایئں اور آخر میں 2 کو i کی Value سے ضرب کیا گیا ہے یاد رہے کے Plus Sign دو چیزوں کو Concatenate یعنی جوڑنے یا ملانے کے کام آتا ہے -

Next | Previous

While Loop

For Loop اور While Loop میں یہ فرق ہے کہ For Loop میں Starting Value اس کے بعد Condition اور آخر میں Ending Value ہوتی ہے اور While Loop میں Condition ہوتی ہے جب تک Condition True ہے Loop چلتا رہے گا جیسے ہی Condition False ہو گی Loop End ہو جائے گا -

```
var i;
while(i <= 10) {
   document.write(i + '<br/>');
   i++;
}
```

آپ دیکھ رہے ہیں کے While Loop کے بعد Condition دی گئی ہے کہ i چھوٹا یا 10 کے برابر ہے جب تک یہ Condition True ہو گی یعنی جب تک i کی 10Value کے برابر نہ ہوگی تب تک While Loop چلتا رہے گا اور Curly Bracket میں وہ کوڈ لکھا گیا ہے جو لوپ میں چلانا ہے پہلی مرتبہ i کی ویلیو 1 ہوگی پھر i میں 1 کا اضافہ ++i سے کیا گیا ہے -

Next | Previous

Reverse While Loop

While Loop کو بھی آپ Reverse میں لکھ سکتے ہیں جیسے آپ نے For Loop کو لکھا تھا تمام Conditions اور Values کو Reverse کر دیں جیسا نیچے دیا گیا ہے -

```
var i = 10;
while(i >= 1) {
   document.write(i + '<br/>');
   i--;  }
```

DoWhile Loop

جیسا کہ آپ کو معلوم ہے While Loop Condition پر چلتا ہے اگر Condition False ہوگی تو لوپ نہیں چلے گا لیکن اگر آپ یہ چاہتے ہیں کے False Condition پر بھی لوپ کم از کم ایک بار ضرور چلے اس کے لیئے آپ کو Do While Loop استعمال کرنا ہوگا اس Loop میں پہلے Do کا Block آتا ہے پھر While کی Condition

آتی ہے اگلے پیج پر False Condition دی گئی ہے لیکن آپ کا
Loop ایک دفعہ ضرور چلے گا کیونکہ اس Loop میں Condition
بعد میں چیک کی جاتی ہے کہ دی گئی True Condition ہے یا
False Condition ہے اگلے پیج پر اس کی مثال دی گئی ہے -

Next | Previous

```
var i = 1;
    do{
      document.write(i + '<br/>');
      i++;
    } while(i >= 10);
```

Break Statement

جیسا کہ نام سے ظاہر ہے Break کا مطلب توڑنا ہے اگر آپ Loop
کو توڑنا چاہتے ہیں یعنی دی گئی Condition سے پہلے ہی Loop
ختم ہو جائے تو اس کے لیئے آپ کو entBreak Statem کو استعمال
کرنا ہو گا -

```
for(i=1; i <= 10; i++)
{
   document.write(i + '<br/>');
   if(i == 5) {
      break;
   }
}
```

جیسے ہی i کی 5Value ہو گی لوپ End ہو جائے گا اور Break
Statement کو آپ While Loop میں بھی استعمال کر سکتے ہیں -

Next | Previous

Continue Statement

Continue Statement کو لوپ میں استعمال کیا جاتا ہے جیسے آپ نے Loop کو ختم کرنے کے لیئے Break Statement کو استعمال کیا تھا اسی طرح سے کسی Specific Condition کو جاری یا Continue رکھنے کے لیے استعمال کرتے ہیں -

```
var i;
for(i=1; i <= 10; i++)
{
   document.write(i + '<br/>');
   if(i == 5)
   {
      document.write('JavaScript' + '<br/>');
      continue;
   }
}
```

جب i کی 5Value کے برابر ہو گی تو JavaScript پرنٹ ہو گا اور پھر Continue Loop چلے گا اور 10 تک گنتی پرنٹ ہو گی -

Next | Previous

Array in JavaScript

آپ جانتے ہیں کہ Variable ایک وقت میں ایک Value ہولڈ یا Store کرتا ہے لیکن اگر آپ ایک سے ذیادہ Value ہولڈ کرنا چاہتے ہیں اس کے لیئے آپ کو Array استعمال کرنا ہوگا Array کو بنانے کے مختلف طریقے ہیں اور اس کو بنانا بہت آسان بھی ہے - Array میں

ہر ویلیو کا ایک ndexI ہوتا ہے اور Index ہمیشہ 0 زیرو سے شروع ہوتا ہے - Array کو بنانے کے لیئے بڑا بریکٹ استعمال کیا جاتا ہے جس میں Array کی ویلیو دی جاتی ہیں Array کو دی گئی مثالوں کی مدد سے سمجھتے ہیں -

```
var a = [ ];
```

اوپر دیا ہوا Array خالی ہے -

```
var a = [1,5,18,25,36,40,45,54];
```

یہ ایک نمبر Array ہے جس میں کوما دے کر ویلیو دی گی ہیں -

```
var a = [1,5,18,'JavaScript',36,54];
```

اوپر دیئے گئے Array میں Mix Data Type ویلیو دی گئی ہے -

Next | Previous

Creating Array

پچھلی مثالوں میں آپ نے دیکھا کہ Array کی ہر Value کو Comma لگا کر لکھا جاتا ہے اور ان Value کو Array کے Element کہتے ہیں اب ان Values کو پرنٹ کر کے دیکھیں گے یاد رہے کہ Array کا Index ہمیشہ 0 صفر سے شروع ہوتا ہے یعنی پہلی ویلیو کا Index0 صفر ہے اور دی گئی مثال میں پہلی ویلیو کو پرنٹ کیا گیا ہے -

```
var a = [2,5,9,15,18,27];
 document.write(a[0]);
```

نیچے دی گئی مثال میں دوسری ویلیو پرنٹ ہو گی یاد رہے کہ جب تک آپ کا براوزر Open ہے Variable یا Array میں ویلیو ہے جیسے ہی آپ کا براوزر بند ہو گا Variable یا Array کی ویلیو ختم ہو جائے گی -

```
    var a = [2,5,9,15,18,27];
    document.write(a[1]);
```

Next | Previous

Looping in Array

Array کے تمام Element کو پرنٹ کرنے کے لیئے Loop استعمال کیا جاتا ہے نیچے دی گئی مثال میں Array کے تمام Element کو ایک ساتھ پرنٹ کیا گیا ہے -

```
    var things = new
Array("chair","table","door","window","wall");
    for(var i=0; i<things.length; i++)
    {
        document.write(things[i] + "<br/>");
    }
```

chair
table
door
window
wall

Length Property کی مدد سے آپ Array میں Values کی تعداد معلوم کر سکتے ہیں - Array میں جتنی Values یا Element ہیں اتنی ہی دفعہ Loop چلے گا اور write فنکشن میں

thing کے اندر بڑے بریکٹ میں [i] کا مطلب Array کا Index ہے اور Array Index کی Value کو دکھاتا ہے اور اس کو ہمیشہ بڑے بریکٹ میں لکھا جاتا ہے -

Next | Previous

Join Function in JavaScript

Array Value کے ساتھ String ملانے کے لیئے Join Function کو استعمال کیا جاتا ہے نیچے دی گئی مثال میں Array کے ساتھ Join Function کو استعمال کیا گیا ہے اور Parameter میں Inverted Commas کے اندر Dash Sign دیا گیا ہے جو ہر Array کی Value کے ساتھ Dash Sign لگا دے گا -

```
    var things = new
Array("chair","table","door","window","wall");
    var total = things.join("-");
    document.write(total);
```

chair-table-door-window-wall

Next | Previous

Sorting an Array

JavaScript میں آپ Array کو Sort بھی کر سکتے ہیں اس کے لیئے Sort Function موجود ہے جو Array کو Alphabeticaly اور Numericaly ترتیب یا Sort کرتا ہے نیچے دی گئی مثال میں

پہلے Array کو Sort کیا گیا ہے پھر Array کی Value کو Loop میں پرنٹ کیا گیا ہے -

```
    var things = new
Array("chair","table","door","apple","wall");
    things.sort();
    for(var i=0; i<things.length; i++)
    {
        document.write(things[i] + " ");
    }
```

apple chair door table wall

یاد رہے کہ آپ صرف Alphabate کو ہی نہیں بلکہ Number Value کو بھی Sort کر سکتے ہیں -

Next | Previous

Reverse Array

Array کی ویلیو کو الٹا یا Reverse لکھنے کے لیئے آپ کو ctionReverse Fun استعمال کرنا ہو گا - یہ فنکشن صرف Array کے Index کو Reverse کرتا ہے یعنی جو ویلیو Zero Index پر ہے وہ آخری Index پر کر دیگا اور جو Value آخری Index پر ہے وہ پہلے Index پر آجائے گی نیچے اس کی مثال دی گئی ہے اور تصویر میں اس کا Output دیا گیا ہے -

```
   var things = new
Array("chair","table","door","apple","wall");
```

```
things.reverse();
for(var i=0; i<things.length; i++)
{
   document.write(things[i] + " ");
}
```

wall apple door table chair

Next | Previous

Searching in Array

JavaScript میں اگر آپ Array میں کسی Value کو Search کرنا چاہتے ہیں اس کے لیئے آپ کو IndexOf Function استعمال کرنا ہو گا - یہ فنکشن آپ کو Value نہیں بلکہ اس Value کا Index بتائے گا کہ یہ Value کس Index پر موجود ہے -

```
var things = new
Array("chair","table","door","apple","wall");
document.write(things.indexOf('door'));
```

اوپر دیئے گئے کوڈ کا جواب 2 آئے گا کیونکہ یہ Value تیسرے Index پر ہے اور IndexOf Function آپ کو اس Value کا Index Return کرے گا -

Length Property

اس Property کی مدد سے آپ Array میں موجود Element کو گن سکتے ہیں دی گئی مثال میں MessageBox میں 5 آئے گا کیونکہ Array میں 5 ویلیو دی گئیں ہیں -

```
    var things = new
Array("chair","table","door","apple","wall");
    document.write(things.length);
```

Next | Previous

Push Function

اگر آپ Array Value کے آخر میں کوئی ویلیو Add کرنا چاہتے ہیں اس کے لیئے Push Function کو استعمال کرنا ہو گا - یاد رہے کہ Push Function آخری Element کے بعد ویلیو کو Add کرتا ہے -

```
  var things = new
Array("chair","table","door","apple","wall");
  things.push('testing');

  for(var i=0; i<things.length; i++)
  {  document.write(things[i] + " ");  }
```

Splice Function

Array سے مخصوص ویلیو کو حاصل کرنے کے لیئے Splice Function کو استعمال کیا جاتا ہے دی گئی مثال میں 2 ویلیو Chair اور Table براوزر پر نظر آیئں گی -

```
    var things = new
Array("chair","table","door","apple","wall");
    things.splice(2);

    for(var i=0; i<things.length; i++)
    { document.write(things[i] + " "); }
```

Next | Previous

Pop Function

Array میں سے آخری ویلیو کو ختم یا Remove کرنے کے لیئے Pop Function کو استعمال کیا جاتا ہے دی گئی تصویر میں آخری Value نہیں ہے -

```
var things = new
Array("chair","table","door","apple","wall");
things.pop();
for(var i=0; i<things.length; i++)
{ document.write(things[i] + " "); }
```

chair table door apple

Unshift Function

آپ نے پچھلی مثال میں Array کے آخر میں ویلیو کو Insert اور Remove کرنا سیکھا تھا اب آپ Array کے شروع میں ویلیو کو Insert اور Remove کرنا سیکھیں گے - Unshift فنکشن سے آپ Array کے شروع میں ویلیو کو Insert کریں گے دی گئی مثال میں Room کو شروع میں Insert یا شامل کیا گیا ہے -

```
var things = new
Array("chair","table","door","apple","wall");
things.unshift('room');
for(var i=0; i<things.length; i++)
{  document.write(things[i] + " ");  }
```

Next | Previous

Shift Function

آپ Shift Function کو استعمال کرتے ہوئے پہلی ویلیو کو Remove یا Delete کر سکتے ہیں نیچے دی گئی مثال کو Apply کر کے چیک کریں -

```
   var things = new
Array("chair","table","door","apple","wall");
   things.shift();
   for(var i=0; i<things.length; i++)
   {
      document.write(things[i] + " ");
   }
```

جب یہ کوڈ Run ہو گا تو آخری ویلیو پرنٹ نہیں ہو گی کیونکہ Shift Function نے آخری ویلیو کو Remove کر دیا ہے -

Next | Previous

Function in JavaScript

آپ نے اب تک ہر کام کو کرنے کے لیئے کوئی نہ کوئی Function استعمال کیا ہے لیکن اب آپ Function کو خود بنانا سیکھیں گے Function کو بنانے کے کچھ Rules ہیں جن کو سمجھنا بہت ضروری ہے - JavaScript کے جو فنکشن آپ نے استعمال کیئے ہیں وہ Built-In Function کہلاتے ہیں اور جو فنکشن آپ خود بنایئں گے وہ Custom Function کہلایئں گے - آیئں اب اس کو بنانا سیکھتے ہیں -

Creating Function

کسی بھی Language میں فنکشن ایک Block of Code ہوتا ہے
جس کو صرف ایک دفعہ لکھا جاتا ہے اور پھر کئی دفعہ استعمال کیا
جاتا ہے یاد رہے کہ Function کو Method بھی کہا جاتا ہے -
فنکشن کو بنانے کے لیئے پہلے Function لکھا جاتا ہے جو کہ ایک
Keyword بھی ہے Keyword ہمیشہ Blue Color میں ہوتے ہیں
اور Keyword کو کبھی بھی Variable Name میں استعمال نہیں کیا
جا سکتا ہے - Function لکھنے کے بعد Function کا نام دیا جاتا
ہے پھر اس کے بعد onFuncti کو ( ) Parenthesis دیتے ہیں اس
کے بعد { } Curly Bracket لگاتے ہیں Curly Bracket کو
Braces بھی کہتے ہیں -

Next | Previous

نیچے دئیے ہوئے کوڈ میں Function بنا کے دکھایا گیا ہے جس کا نام
printText ہے یاد رہے کہ فنکشن کو بنانے کے بعد اس کو استعمال
کرنے کے لیئے اس کو Call کیا جاتا ہے - فنکشن کو Call کرنے
کے لیئے صرف فنکشن کا نام لکھتے ہیں اس کے بعد Parenthesis
لگاتے ہیں آخر میں Semi Colon تے لگا ہیں جیسا نیچے دیا گیا ہے
-

```
function printText()
{
```

```
    document.write("This is a function" + "<br/>");
}

printText();
```

فنکشن کو کال کرنے کے بعد آپ دیکھیں گے کہ براوزر پر Text Print ہو گا اب اسی فنکشن کو آپ 3 دفعہ کال کریں تو یہ 3 دفعہ Text Print کرے گا - امید ہے کہ آپ کو فنکشن سمجھ میں آ گیا ہو گا -

Next | Previous

اب ایک ایسا فنکشن بناتے ہیں جو دو نمبروں کو جمع کرتا ہو نیچے اس کی مثال دی گئی ہے اب اس فنکشن کو کال کریں یہ آپ کو 9 جمع کر کے بتائے گا لیکن یہ Fix فنکشن ہے جتنی بار آپ اس کو کال کریں گے یہ آپ کو ایک ہی جواب دے گا -

```
function doSum()
{
    document.write(5+4);
}
```

Function Parameter

اگر آپ یہ چاہتے ہیں کہ Function کو جتنی بار کال کیا جائے یہ آپ کو مختلف Values کو جمع کر کے بتائے اس کے لیئے آپ کو ہر بار فنکشن کو الگ الگ Values دینی ہو گی - فنکشن کو ویلیو فنکشن

کے Parenthesis کے اندر پاس کی جاتی ہے یہ ویلیو فنکشن کے
Parameter کھلاتی ہیں اب اس کو مثال سے سمجھتے ہیں -

```
function doSum(num1 , num2)
{
    document.write(num1 + num2 + '<br/>');
}
```

Next | Previous

پچھلی مثال میں دو ویلیو 1num اور 2num کو فنکشن کے
Parameter میں پاس کیا گیا ہے Comma لگا کر آپ جتنے چاہیں
Parameter یا Values پاس کر سکتے ہیں - Parameter کے نام
کچھ بھی ہو سکتے ہیں آپ چاہیں تو اپنا نام بھی لکھ سکتے ہیں لیکن
Parameter کا نام Meaningfull ہونا ضروری ہے اب اس فنکشن
کو کال کریں اور اس کے Parameter میں 2 ویلیو کو پاس کریں -

```
function doSum(num1 , num2)
{
   document.write(num1 + num2);
}

doSum(4,6);
doSum(25,7);
doSum(1,30);
```

آپ نے اس فنکشن کو 3 دفعہ کال کیا ہے اور ہر دفعہ ایک نئی ویلیو
دی ہے جو ویلیو آپ نے Parameter میں پاس کی ہے وہ ویلیو آگے
فنکشن کے Parameter یعنی 1num اور 2num میں آگئی ہے اور

Function Block میں write.document کے اندر جمع ہو کر پرنٹ ہوگئی ہے -

Next | Previous

ویلیو پاس کرتے وقت بھی آپ نے Comma لگا کر ویلیو کو پاس کرنا ہے فنکشن بناتے وقت وہی اصول کو اپلائی کرنا ہو گا جو Variable بناتے وقت Follow کرتے ہیں یاد رکھیں کہ فنکشن کے Parameter کو Argument یا Signature بھی کہتے ھیں -

Function Return Value

آپ ایک ایسا فنکشن بھی بنا سکتے ہیں جو ویلیو کو Return بھی کرتا ہو اس کے لئے آپ کو eturn StatementR کو استعمال کرنا ہوگا جیسا کہ نام سے ظاھر ھے Return کا مطلب واپس کے ہیں یعنی ایک ایسا فنکشن جو ویلیو کو Return کرتا ہے دی گئی مثال میں 2 نمبروں کو جمع کرنے کے بعد Value کو فنکشن سے Return کیا گیا ہے اور پھر ایک ویری ایبل ans کے اندر اس کو ldHo کیا گیا ہے جب Variable کے اندر ویلیو آگئی تو اس کو write.document سے پرنٹ کیا گیا ہے -

```
function doSum(num1 , num2)
{
   return(num1 + num2);
}
var ans = doSum(24,5);
```

```
document.write(ans);
```

Next | Previous

Confirm Dialog

بعض اوقات یوزر سے کسی کام کو کرنے کے لیئے اجازت لیتے ہیں
مثال کے طور پر اگر آپ کسی Recorde کو Delete کرنا چاہتے ہیں
تو اس کے لیئے آپ کو پہلے یوزر سے Confirm کرنا ہوگا جیسے
Are you sure you want to delete that record اگر یوزر Yes
کرے تو پھر Recorde Delete ہو ورنا نہیں اور اگر No کرے تو
کچھ بھی نہیں ہو اس قسم کے کاموں کو کرنے کے لیئے Confirm
Dialog کو استعمال کرتے ہیں اب اس کی مثال دیکھتے ہیں -

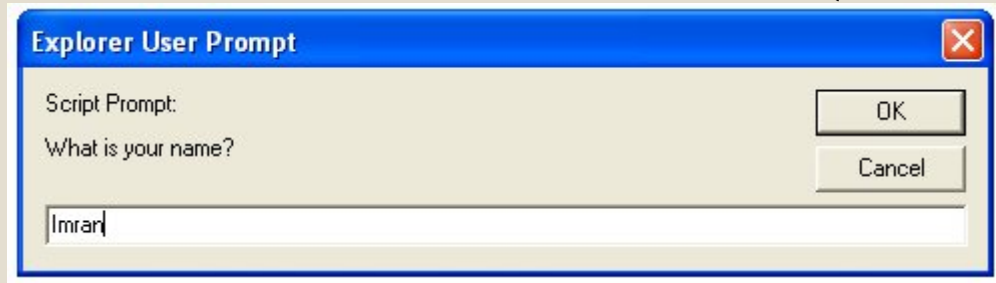
```
if(confirm("Are you sure you want to do that?"))
{
     alert("You said yes");
}
else
{
    alert("You said no");
}
```

ایک بات یاد رہے کہ Confirm Dialog آپ کو True یا False
Return کرے گا اگر آپ Yes کریں گے تو True اور اگر No کریں
گے تو False Return ہوگا -

Next | Previous

پچھلی مثال میں اگر آپ Yes کریں گے تو if کا اسٹیٹمنٹ چلے گا اور اگر No کریں گے تو Else Statement چلے گا -

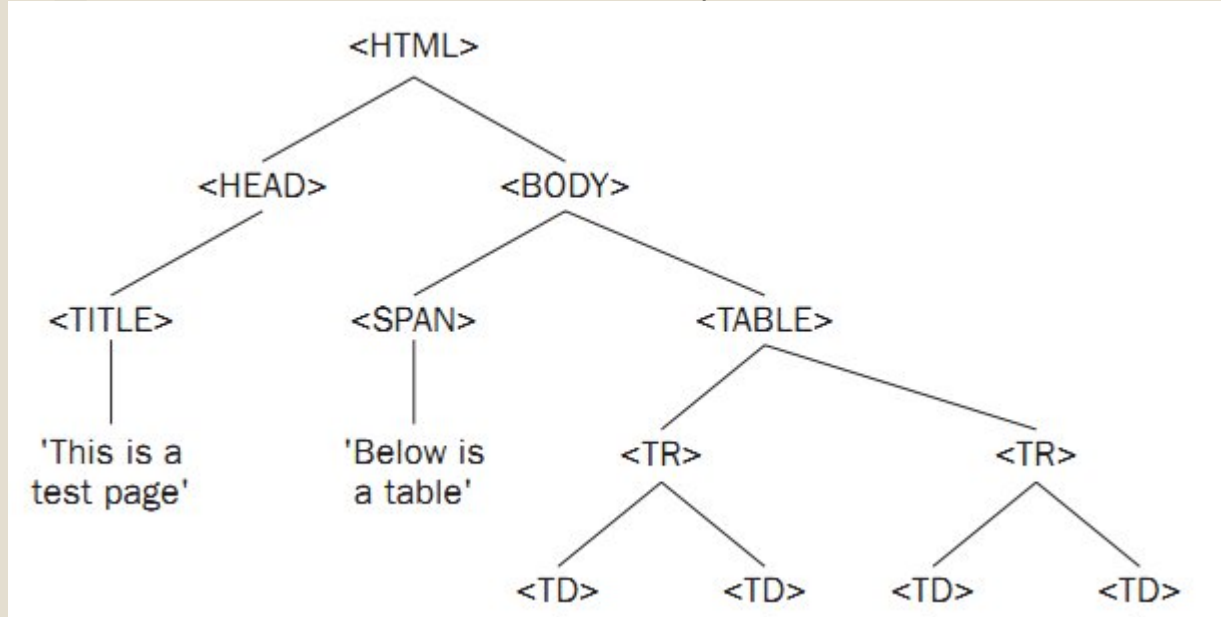
Prompt in JavaScript

User سے Input حاصل کرنے کے لیئے Prompt Dialog کو Use کیا جاتا ہے یہ بھی پچھلی مثال کی طرح True یا False Return کرتا ہے اگر آپ Ok کریں گے تو True Return ہو گا اور اگر Cancel کریں گے تو False Return ہو گا اور Ok پر if کا کوڈ چلے گا اور Cancel پر else والا کوڈ Execute ہو گا -

```
var ans = prompt("What is your name?" , "");
if(ans)
{  alert("My name is " + ans);  }
else
{  alert("You refused to answer");  }
```

Next | Previous

Document Object Model

یہ JavaScript کا سب سے اهم Topic ہے اس کو سمجھنا بہت ضروری ہے اور اس کو سمجھنا آسان بھی ہے آپ کو HTML آتی ہے دراصل HTML کی ترتیب کو ہی DOM کہتے ہیں جیسے سب سے پہلے آپ کو Doctype Tag نظر آتا ہے اس کے بعد HTML Tag پھر Head Tag اور اس کے اندر Title, Link, Script, Style اور Meta Tag ہوتے ہیں - Tag کے اندر Sub Tag کو Heirarchy بھی کھتے ہیں اس کے بعد Body Tag آتا ہے اور باقی تمام ٹیگ Body Tag کے اندر ہوتے ہیں اور یہ Body Tag کی Heirarchy کھلاتی ہے - جاوا اسکرپٹ میں مکمل Ability ہے تمام چیزوں کو تبدیل کرنے کی اور اس Technic کو DHTML یعنی Dynamic HTML کہتے ہیں - Document Object Model دراصل ایک Standard ہے جو World Wide Web Consortium3W C نے بنایا ہے اور DOM Developer کو ایک راستہ فراہم کرتا ہے جس کی مدد سے آپ Web Page پر موجود ہر چیز کو Access کر سکتے ہیں چند Property اور Method کو استعمال کرتے ہوئے جیسے Image, Table, Text اور بہت کچھ آپ آگے کی مثالوں میں دیکھیں گے -

Document Object Model کو سمجھنے کے لیئے نیچے دی گئی تصویر کو دیکھیں جس میں Heirarchy یا Tree دکھلائی گئی ہے -

امید ہے کہ آپ کو Document Object Model سمجھ میں آ گیا ہو گا اس کی پہلی مثال اگلے پیج پر ملاحضہ فرمائیں -

Next | Previous

getElementById Function

JavaScript میں HTML کے کسی بھی Tag کو حاصل کرنے کے لیئے اس فنکشن کو استعمال کرتے ہیں - اس Function کا کام کسی بھی HTML Element کو Id کے زریعہ سے اس Element کو حاصل کرنا ہے 1H Tag کے اندر Id کا نام myheading دیا گیا ہے - نیچے دی گئی مثال میں Body Tag کے اندر 1H Heading کے نیچے JavaScript کا کوڈ لکھیں -

```
<h1 id="myheading">My Heading</h1>


<script type="text/javascript">
    var heading =
document.getElementById('myheading').innerHTML;
    alert(heading);
</script>
```

JavaScript کوڈ میں آپ ایک نئی Property دیکھ رہے ہیں جس کا نام innerHTML ہے اس کا کام دی گئی Id کا Text حاصل کرنا یا تبدیل کرنا ہے مثال میں Text کو حاصل کر کے ایک Variable میں اسٹور کیا گیا ہے اور پھر اس کو Message Box میں دکھایا گیا ہے -

Next | Previous

آگے جاکر آہستہ آہستہ مثالیں تھوڑی سی مشکل ہوتی چلی جائیں گی اب دوسری مثال دیکھیں گے جس میں فونٹ کا کلر تبدیل کیا گیا ہے -

```
<h1 id="myheading">This is heading1</h1>
```

```
var h1Element = document.getElementById('myheading');
h1Element.style.color = 'red';
```

پہلی لائین میں Element کو حاصل کیا گیا ہے اس کے بعد دوسری لائین میں فونٹ کلر تبدیل کیا گیا ہے اب آپ اس کو براوزر پر چلائیں گے تو Heading کا کلر تبدیل ہو گیا ہو گا -

getElementByTagName Function

اب آپ ٹیبل کا کلر تبدیل کریں گے اس کے لیئے آپ کو پہلے ٹیبل بنانا ہو گا اس کے بعد ٹیبل کے نیچے جاوا اسکرپٹ کا کوڈ لکھنا ہو گا آیئے اب اس کی پریکٹس کرتے ہیں -

Next | Previous

```
<table border="1px" width="250px" cellspacing="0px">
  <tr>
  <td>Employee Id</td>
  <td>Employee Name</td>
  </tr>
  <tr>
  <td>12045</td>
  <td>Salman Khan</td>
  </tr>
  </table>


  var tdElement = document.getElementsByTagName('td');
  var tdLength = tdElement.length;

  for(var i = 0; i < tdLength; i++)
  {
```

```
    tdElement[i].style.color = 'gray';
  }
```

Employee Id	Employee Name
12045	Salman Khan

Next | Previous

اس مثال میں آپ کو ایک نیا فنکشن سیکھنے کو ملا ہے جس کا نام getElementsBy TagName ہے اس کو استعمال کرتے ہوئے آپ پیج پر موجود کسی بھی ٹیگ کو پورے پیج پر جتنی دفعہ لکھا گیا ہے حاصل کر سکتے ہیں جیسا کہ آپ نے دیکھا ہے کے ٹیبل میں موجود تمام td کو اس فنکشن کی مدد سے Select کیا گیا ہے پھر اس کو Count کیا گیا ہے Length Property کی مدد سے اس کے بعد ایک لوپ چلایا گیا ہے جو 0 سے شروع ہو کر td کی تعداد تک چلا ہے - صفر 0 سے مراد سب سے پہلا td ہے اس طرح سے سب سے پہلے پہلے td کا کلر پھر دوسرے td کا کلر اور اس طرح سے تمام td کا کلر تبدیل ہو جائے گا لوپ کو استعمال کرتے ہوئے - اگر آپ کو سمجھنے میں کوئی پریشانی ہو تو ایک بار دوبارہ سے اس کو پڑھ لیجیئے سمجھ میں آ جائے گا -

setAttribute Function

اب آپ HTML Element Attribute کو اپلائی کرنا سیکھیں گے اس کے لیئے آپ کو ایک پیراگراف لینا ہوگا اور JavaScript کی مدد سے اس کو سینٹر میں کریں گے -

```
<p id="mypara">This is simple paragraph.</p>
var pElement = document.getElementById('mypara');
pElement.setAttribute('align', 'center');
```

Next | Previous

آپ ہر مثال میں ایک نئی چیز سیکھ رہے ہیں اور آگے بڑھ رہے ہیں اس مثال میں آپ نے setAttribute فنکشن کو سیکھا ہے یہ دو پیرامیٹر لیتا ہے پہلا پیرامیٹر Attribute ہے اور دوسرا پیرامیٹر Value ہے پچھلی مثال میں Align Attribute ہے اور Value Centre دی گئی ہے -

getAttribute Function

پچھلی مثال میں آپ نے Attribute کو Set کرنا سیکھا تھا اب Attribute کو Get یعنی حاصل کرنا سیکھیں گے اس مثال میں Paragraph کو Right Align کیا گیا ہے اور پھر JavaScript کو استعمال کرتے ہوئے Align Attribute کی Value کو حاصل کریں گے -

```
<p id="mypara" align="right">This is simple
paragraph.</p>
```

اب اس کے نیچے جاوا اسکرپٹ کا کوڈ لکھیں -

```
    var pElement = document.getElementById('mypara');
    alert(pElement.getAttribute('align'));
```

Next | Previous

آپ دیکھ رہے ہیں کے پہلی لائن میں Element کو حاصل کیا گیا ہے پھر دوسری لائن میں getAttribute کو استعمال کرتے ہوئے Attribute کا نام دیا گیا ہے جب یہ کوڈ چلے گا تو الرٹ میں Attribute کی ویلیو Right آئے گی -

removeAttribute Function

اب تک آپ نے جاوا اسکرپٹ میں Attribute کو Get اور Set کرنا سیکھا تھا لیکن اب آپ Attribute کو Delete یا Remove کرنا سیکھیں گے ایک پیراگراف کو Right Align کیا گیا ہے جب Page Load ہو گا تو پیراگراف Right Align ہو گا پھر ایک MessageBox آئے گا اور اس کے بعد Attribute Remove ہو جائے گا اور پیراگراف کی Alignement ختم ہو جائے گی یاد رہے کہ ByDefault پیراگراف Left Align ہوتا ہے -

```
    <p id="mypara" align="right">This is simple
paragraph.</p>

    var pElement = document.getElementById('mypara');
    alert('Attribute is going to remove');
```

```
    pElement.removeAttribute('align');
```

Next | Previous

createElement Function

JavaScript اتنی Powerful ہے کہ آپ JavaScript میں New HTML Element بھی بنا سکتے ہیں اس کے لیئے آپ کو اس فنکشن کو استعمال کرنا ہوگا نیچے دی گئی مثال میں 1H Tag Create ہوچکا ہے اس طریقے کو Dynamic Element یا Dynamic Tag کہتے ہیں -

```
 var newElem = document.createElement('h1');
```

createTextNode Function

JavaScript میں Dynamic Element کو نہ صرف Create کرسکتے ہیں بلکہ اس کو Text بھی دے سکتے ہیں اس فنکشن کی مدد سے آپ Dynamic HTML Tag کو Text دینگے اس کی مکمل مثال آگے دی گئی ہے -

```
    var newText = document.createTextNode('This is my
heading');
```

Next | Previous

appendChild Function

پچھلی دونوں مثالوں میں Dynamic HTML Tag کو بنانا اور Text دینے کے فنکشن بتائے تھے اس مثال میں Dynamic Tag کو Body

Tag میں شامل کرنا سیکھایا گیا ہے پہلی دو لایئنیں آپ پچھلی مثال میں دیکھ چکے ہیں پہلی لائن میں Element بنانے کی بات ہو رہی ہے اور دوسری لائن میں Element کا Text بتایا جا رہا ہے - ایک نیا فنکشن appendChild ہے یہ کسی بھی Element کے اندر دوسرا Element شامل کرنے کے لیئے استعمال کیا جاتا ہے تیسری لایئن میں Element کے Variable کے ساتھ appendChild کو استعمال کیا گیا ہے اور اس کے اندر Text کا Variable دیا گیا ہے یہ 1H Element کا Text ہے اب Element مکمل ہو گیا ہے چوتھی لایئن میں اس Element کو Body Tag کے اندر شامل کیا گیا ہے تاکہ آپ اس کو براوزر پر دیکھ سکیں -

```
    var newElem = document.createElement('h1');
    var newText = document.createTextNode('This is my
heading');

    newElem.appendChild(newText);
    document.body.appendChild(newElem);
```

Next | Previous

آپ نے دیکھا کے appendChild فنکشن تو ایک ہی ہے لیکن اس کا استعمال دونوں جگہ پر مختلف ہے تیسری لائن میں یہ 1H Element کا Text شامل کر رہا ہے اور چوتھی لائن میں یہ Body Tag میں

H1 Element کو شامل کر رہا ہے امید ہے کہ آپ کو یہ مثال سمجھ میں آ گئی ہو گی -

fontStyle Property

JavaScript میں آپ CSS کا کام بھی کر سکتے ہیں نیچے دی گئی مثال میں ٹیکسٹ کو Italic کیا گیا ہے Style Property کے اندر fontStyle Property موجود ہے جس کی ویلیو Italic دی گئی ہے اب اس کوڈ کو براوزر پر چلایئں آپ کو ٹیکسٹ Italic نظر آئے گا -

```
<h1 id="myheading">This is heading1</h1>
var myh1 = document.getElementById('myheading');
myh1.style.fontStyle = 'italic';
```

Next | Previous

CSS Class in JavaScript

جاوا اسکرپٹ میں آپ CSS کی Class کو بھی اپلائی کر سکتے ہیں پہلے آپ کو CSS Class کو بنانا ہو گا جیسا کہ نیچے mystyle Class بنائی گئی ہے -

```
<style type="text/css">
.myStyle
{
    font: 18pt arial italic;
    text-decoration: underline;
}
</style>
```

اب آپ Body Tag کے اندر Paragraph کو بنایئں -

```
<p id="mypara">This is simple paragraph.</p>
```

اس کے بعد آپ Property classname کو لکھیں اور اس کی ویلیو CSS Class کا نام Inverted Commas میں لکھیں Class Apply ہو جائے گی -

```
var para = document.getElementById('mypara');
para.className = 'myStyle';
```

Next | Previous

Date Object

JavaScript میں آپ Current Time اور Date کو بھی حاصل کر سکتے ہیں اس کے لیئے آپ کو Date Object کو استعمال کرنا ہو گا جاوا اسکرپٹ میں چند Object موجود ہیں اور آپ اپنا خود بھی Object بنا سکتے ہیں Object دو چیزوں پر مشتمل ہوتا ہے ایک Property اور دوسرا Method یا Function ہے آیئے دیکھتے ہیں کہ اس کو کس طرح استعمال کیا جاتا ہے -

```
var d = new Date();
var mytime = d.getHours();
document.write(mytime);
```

سب سے پہلے Variable لیا گیا ہے اور Date Object سے پہلے New Keyword کو استعمال کیا گیا ہے New کے معنی ہیں کہ

Date Object میں جو Property اور Method موجود ہیں ان تمام کو Variable d میں کاپی کرنا یا شامل کرنا ہے اس کے بعد دوسری لائئن میں ویری ایبل کے ساتھ فنکشن getHours کو استعمال کیا گیا ہے - یہ فنکشن آپ کو Current Hours بتاتا ہے جس کو دوسرے ویری ایبل میں لے کر براوزر پر پرنٹ کیا گیا ہے -

Next | Previous

getDay Function

تاریخ کے علاوہ آپ دنوں Days کو بھی حاصل کر سکتے ہیں اس کے لیئے آپ کو getDay فنکشن کو استعمال کرنا ہو گا یاد رہے کہ آپ کو دن ہمیشہ نمبر میں حاصل ہو گا Sunday کا نمبر Zero صفر ہے اور Monday کا نمبر 1 ہے اس طرح سے پورے ہفتے کا نمبر 0 سے 6 تک ہے اگر آپ نمبر کی جگہ دن کا نام دیکھنا چاہتے ہیں تو آپ کو نیچے والا کوڈ استعمال کرنا ہو گا -

```
var d = new Date();
var myday = d.getDay();
document.write(myday);
```

اوپر والے کوڈ سے آپ کو نمبر حاصل ہوگا اور نیچے والے سے دن کا نام بتائے گا بقیہ کوڈ اگلے پیج پر ملاحضہ فرمایئں -

```
var d = new Date();
var myday = d.getDay();
if(myday == 0)
{
```

```
        document.write('Sunday');
     }
```

Next | Previous

```
else if (myday == 1)
    {
       document.write('Monday');
    }
    else if (myday == 2)
    {
       document.write('Tuesday');
    }
    else if (myday == 3)
    {
       document.write('Wednesday');
    }
    else if (myday == 4)
    {
       document.write('Thursday');
    }
    else if (myday == 5)
    {
       document.write('Friday');
    }
    else
    {  document.write('Saturday');  }
```

آپ نے دیکھا کہ لوجک لگا کر نمبر کے بدلے دن کے نام کو حاصل کیا گیا ہے -

Next | Previous

getDate Function

Current Date حاصل کرنے کے لیئے getDate فنکشن کو استعمال کرتے ہیں -

```
var d = new Date();
var mydate = d.getDate();
document.write(mydate);
```

getFullYear Function

GetFullYear فنکشن آپ کو Current سال بتائے گا -

```
var d = new Date();
var myyear = d.getFullYear();
document.write(myyear);
```

getMonth Function

اس فنکشن کی مدد سے آپ کرنٹ مہینہ حاصل کر سکتے ہیں لیکن یہ پہلے مہینے کو صفر 0 سے شروع کرتا ہے اس طرح سے آخری مہنیہ کا نمبر 11 ہو گا -

```
var d = new Date();
var mymonth = d.getMonth();
document.write(mymonth);
```

Next | Previous

toDateString Function

اس فنکشن کو استعمال کرتے ہوئے آپ کرنٹ تاریخ کو لکھ سکتے ہیں - تاریخ کے ساتھ یہ آپ کو دن بھی دے گا -

```
var d = new Date();
var mydate = d.toDateString();
```

```
document.write(mydate);
```

toLocaleDateString Function

اس فنکشن سے آپ مکمل دن اور مکمل مہینہ لکھ سکتے ہیں - نیچے اس کی مثال دی گئی ہے -

```
var d = new Date();
var mydate = d.toLocaleDateString();
document.write(mydate);
```

Next | Previous

toLocaleTimeString Function

جاوا اسکرپٹ میں toLocalTimeString فنکشن کی مدد سے آپ کرنٹ ٹائم حاصل کر سکتے ہیں -

```
var d = new Date();
var mytime = d.toLocaleTimeString();
document.write(mytime);
```

setDate Function

اب تک آپ نے ڈیٹ کو حاصل کرنا سیکھا ہے اب اس کو تبدیل کرنا سیکھیں گے - نیچے دی گئی مثال میں ڈیٹ اوبجیکٹ کو ایک تاریخ دی گئی ہے جو کہ 5 اگست ہے پھر اس کو تبدیل کیا گیا ہے اور اب تاریخ 9 اگست ہو گئی ہے -

```
var dt = new Date("Aug 5 2013");
dt.setDate(9);
document.write(dt);
```

Next | Previous

setFullYear Function

اس فنکشن کی مدد سے آپ موجودہ سال کو تبدیل کر سکتے ہیں پہلے تاریخ کو حاصل کیا گیا ہے پھر اس میں 1 کو جمع کر کے سال کو تبدیل کیا گیا ہے -

```
var d = new Date("Aug 5 2013")
d.setFullYear(d.getFullYear() + 1)
document.write(d)
```

setHours Function

اس فنکشن کو استعمال کرتے ہوئے آپ ٹائم کو بھی تبدیل کر سکتے ہیں -

```
var dt = new Date("Aug 28, 2010 03:00:00");
dt.setHours(5);
document.write(dt);
```

setMinutes Function

setMinutes فنکشن سے آپ Minutes کو بھی تبدیل کر سکتے ہیں -

```
var dt = new Date("Aug 28, 2010 03:27:00");
dt.setMinutes(45);
document.write(dt);
```

Next | Previous

setMonth Function

نیچے دی گئی مثال میں مہینے کو تبدیل کیا گیا ہے -

```
var dt = new Date("Aug 28, 2010");
dt.setMonth(8);
document.write(dt);
```

Browser Name

JavaScript میں آپ اپنے Browser کا نام بھی معلوم کر سکتے ہیں اس کے لیئے آپ کو navigator.appName Property کو استعمال کرنا ہوگا ۔

```
document.write(navigator.appName);
```

Browser Version

نیچے دیئے گئے کوڈ سے آپ اپنے براوزر کا ورژن دیکھ سکتے ہیں ۔

```
document.write(navigator.appVersion);
```

Next | Previous

String Function

JavaScript میں String کے بہت سے Function ہیں جیسے String Value کو Count کرنا یا Bold کرنا یا Search کرنا وغیرہ شامل ہیں ۔ آپ جانتے ہیں کہ Variable ایک وقت میں ایک ویلیو اسٹور کرتا ہے لیکن اگر آپ چاھتے ہیں کہ ایک Variable پچھلی ویلیو کو بھی اسٹور کرے تو اس کے لیئے آپ کو نیچے دی گئی مثال کو سمجھنا ہو گا ۔

```
var msg = "This is";
msg = msg + " String Object";
msg = msg + " in JavaScript.";
document.write(msg);
```

دوسری لائن میں Variable میں پہلی لایئن کی ویلیو Store ہے اس کو Plus Sign سے دوسری لایئن کی نئی ویلیو کو جمع کرکے دوبارہ اسی Variable میں ویلیو Store کی گئی ہے تیسری لایئن میں بھی ایسا ہی کیا گیا ہے اس طرح سے تینوں لائینوں کی ویلیو ایک ہی Variable میں جمع یا Concatenate ہو جایئں گی اس کو لکھنے کا دوسرا طریقہ اگلے پیج پر ملاحضہ فرمایئں -

Next | Previous

یہ شارٹ طریقہ ہے پچھلی ویلیو میں نئی ویلیو کو شامل کرنے کا فرق صرف اتنا ہے کہ Variable کا نام دوبارہ نہیں لکھا گیا ہے بلکہ Plus Sign کو = Equal سے پہلے لگایا گیا ہے اس کا مطلب بھی یہی ہے کہ پچھلی ویلیو میں اگلی ویلیو جمع کر دی جائے -

```
var msg = "This is";
msg += " String Object";
msg += " in JavaScript.";
document.write(msg);
```

This is String Object in JavaScript.

toUpperCase Function

Variable کے تمام Alphabate کو Capital Letter میں تبدیل کرنے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے -

```
var msg = "This is String Object";
var bigwords = msg.toUpperCase();
document.write(bigwords);
```

Next | Previous

پچھلی مثال کو آپ اس طرح بھی لکھ سکتے ہیں فرق صرف یہ ہے کہ پہلے آپ نے ایک Variable لیا پھر اس کو پرنٹ کیا تھا اب ڈائیریکٹ فنکشن براوزر پر پرنٹ ہوگا -

```
var msg = "This is String Object";
document.write(msg.toUpperCase());
```

toLowerCase Function

اس فنکشن سے آپ تمام Alphabate کو چھوٹے حروف میں تبدیل کر سکتے ہیں -

```
var msg = "This is String Object";
document.write(msg.toLowerCase());
```

Length Property

Variable ویلیو کے Character کو Count کرنے کے لیئے Length Property کو استعمال کرتے ہیں دی گئی ویلیو کی تعداد 14 ہے اس میں Blank Space بھی شامل ہے یعنی Blank Space بھی ایک Character ہے -

```
var msg = "This is String";
document.write(msg.length);
```

Next | Previous

big Function

JavaScript میں اس فنکشن کی مدد سے آپ ٹیکسٹ کو بڑا کر سکتے ہیں -

```
var msg = "This is String";
document.write(msg.big());
```

bold Function

Bold Function ٹیکسٹ کو بولڈ کرنے کے لیئے استعمال ہوتا ہے -

```
var msg = "This is String";
document.write(msg.bold());
```

charAt Function

کونسا Character کس Index پر ہے یہ معلوم کرنے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے یہ ایک پیرامیٹر لیتا ہے نیچے اس کی مثال کو دیکھیں -

```
var msg = "This is String";
document.write(msg.charAt(0));
```

Next | Previous

جیسا کہ آپ دیکھ رہے ہیں کے charAt فنکشن کے پیرامیٹر میں 0 صفر ویلیو دی گئی ہے اس کا مطلب پہلا Character ہے یہ براوزر

پر آپ کو T دکھائے گا پیرامیٹر میں جو نمبر لکھیں گے یہ آپ کو اسی Index کا ٹیکسٹ لا کر دے گا -

charCodeAt Function

آپ جانتے ہیں کہ ہر Key کا ایک کوڈ ہوتا ہے اس کو ASCII Code ایسکی کوڈ کہتے ہیں جب آپ اپنے Keyboard سے کوئی بھی Key کو Press کرتے ہیں تو وہ Convert ہوتا ہے ASCII Code میں پھر اس کے بعد ASCII Code کو Convert کیا جاتا ہے 0 یا 1 میں اس طرح سے ہر Key کا ایک الگ کوڈ ہے charCodeAt فنکشن کی مدد سے آپ کسی بھی Character کا کوڈ دیکھ سکتے ہیں دی گئی مثال میں T کا ASCII Code بتائے گا جو کے 84 ہے -

```
var msg = "This is String";
document.write(msg.charCodeAt(0));
```

Next | Previous

fontcolor Function

ٹیکسٹ کا کلر تبدیل کرنے کے لیئے آپ اس فنکشن کو استعمال کرینگے اس مثال میں ویلیو کا Color Red ہو جائے گا -

```
var msg = "This is JavaScript";
document.write(msg.fontcolor('red'));
```

italics Function

جاوا اسکرپٹ میں آپ ٹیکسٹ کو اٹالک بھی کر سکتے ہیں -

```
var msg = "This is JavaScript";
document.write(msg.italics());
```

toString Function

کسی بھی نمبر ویلیو کو اسٹرنگ میں تبدیل کرنے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے اس Variable کی ڈیٹا ٹائپ اب Number سے String ہو چکی ہے -

```
var msg = 27;
document.write(msg.toString());
```

Next | Previous

concat Function

ایک Variable کو دوسرے Variable سے ملانے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے پہلے ویری ایبل کے ساتھ اس فنکشن کے اندر دوسرے ویری ایبل کو دیا گیا ہے اب پہلے ویری ایبل کی ویلیو دوسرے ویری ایبل کی ویلیو کے ساتھ مل گئی ہے -

```
var str1 = "Java ";
var str2 = "Script!";
document.write(str1.concat(str2));
```

fontsize Function

فونٹ کا سائز کم یا زیادہ کرنے کے لئے اس فنکشن کو یوز کرتے ہیں - اس کا سائز 1 سے 7 تک ہے سب سے چھوٹے سائز کے لیئے 1 اور سب سے بڑے سائز کے لیئے 7 کو استعمال کرتے ہیں -

```
var mystr = "Java Script";
document.write(mystr.fontsize(7));
```

Next | Previous

indexOf Function

یہ معلوم کرنے کے لیئے کہ کونسا کریکٹر کس انڈیکس پر ہے اس فنکشن کو استعمال کرتے ہیں اور انڈیکس ہمیشہ 0 صفر سے شروع ہوتا ہے - وی کا انڈیکس 2 ہے -

```
var mystr = "Java Script";
document.write(mystr.indexOf("v"));
```

آپ نہ صرف ایک کریکٹر بلکہ ایک ورڈ کا Index بھی معلوم کر سکتے ہیں اس کی مثال نیچے دی گئی ہے -

```
var mystr = "Java Script";
document.write(mystr.indexOf("Script"));
```

lastIndexOf Function

اس فنکشن کے پیرامیٹر میں آپ صرف ویلیو کا آخری الفاظ دیں گے اور یہ آپ کو اس ورڈ کا انڈیکس بتائے گا -

```
var mystr = "Welcome to Java Script";
document.write(mystr.lastIndexOf("Script"));
```

Next | Previous

replace Function

کسی بھی ورڈ کو دوسرے ورڈ سے تبدیل کرنے کے لیئے replace فنکشن کو استعمال کیا جاتا ہے یہ فنکشن دو پیرامیٹر لیتا ہے پہلا پیرامیٹر میں وہ ورڈ آتا ہے جس کو تبدیل کرنا ہو اور دوسرا پیرامیٹر وہ ہے جس سے تبدیل کیا جا رہا ہے دی گئی مثال میں script کو programing سے تبدیل کیا جا رہا ہے -

```
var mystr = "Welcome to Java Script";
document.write(mystr.replace("Script","Programing"));
```

search Function

JavaScript میں آپ کسی بھی ورڈ کو سرچ کر سکتے ہیں اگر ویلیو موجود ہو گی تو یہ آپ کو اس ویلیو کا Index بتائے گا اور اگر ویلیو موجود نہیں ہوگی تو یہ آپ کو منفی 1- ویلیو ریٹرن کرے گا نیچے کی مثال میں Script کا Index معلوم کیا جا رہا ہے -

```
var mystr = "Welcome to Java Script";
document.write(mystr.search("Script"));
```

Next | Previous

slice Function

اسٹرنگ ویلیو سے چند کریکٹر کو حاصل کرنے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے اس فنکشن کے پیرامیٹر میں آپ جو نمبر دیں گے اتنے ہی کریکٹر اسٹرنگ ویلیو سے نہیں دکھائے گا دی گئی مثال

میں 2 کریکٹر کے بعد سے تمام کریکٹر آپ کو Browser پر Show ہونگے -

```
var mystr = "JavaScript";
document.write(mystr.slice(2));
```

اس کی دوسری مثال میں 2 پیرامیٹر دیئے گئے ہیں پہلا پیرامیٹر کہاں سے کریکٹر دکھانا ہے اور دوسرا کہاں تک دکھانا ہے دی گئی مثال میں ویلیو سے 2 کریکٹر کے بعد سے 5 کریکٹر تک دکھانا ہے نیچے دی گئی مثال کے جواب میں صرف vas آپ کو براوزر پر نظر آئے گا کیونکہ s پانچوے نمبر پر ہے -

```
var mystr = "JavaScript";
document.write(mystr.slice(2,5));
```

Next | Previous

اس فنکشن کے پیرامیٹر میں منفی ویلیو دینے سے آخری کریکٹر سے دکھانا شروع کرتا ہے دی گئی مثال میں آپ دیکھ سکتے ہیں براوزر پر آخری دو کریکٹر pt نظر آیئں گے باقی الفاظ نظر نہیں آیئں گے -

```
var mystr = "JavaScript";
document.write(mystr.slice(-2));
```

small Function

Small Function کے استعمال سے الفاظوں کا Size چھوٹا ہو جاتا ہے -

```
var mystr = "JavaScript";
```

```
document.write(mystr.small());
```

split Function

String Values میں سے الفاظوں کو توڑنے یا الگ الگ کرنے کے لیئے split فنکشن کو استعمال کیا جاتا ہے اور یہ فنکشن ورڈ کو الگ کرکے ایک Array بنا دیتا ہے اگلے پیج پر اس کی مثال دیکھتے ہیں -

Next | Previous

جب آپ اس کو براوزر پر دیکھیں گے تو اسپیس کی جگہ پر آپ کو کوما نظر آئے گا یہ فنکشن اسپیس کو کوما میں تبدیل کردیتا ہے -

```
var str="Welcome to JavaScript";
document.write(str.split(" "));
```

Welcome,to,JavaScript

strike Function

یہ فنکشن الفاظوں کے اوپر بیچ میں ایک لائن لگا دیتا ہے -

```
var str="Welcome to JavaScript";
document.write(str.strike());
```

~~Welcome to JavaScript~~

sup Function

یہ فنکشن ورڈ یا کریکٹر کو لائن سے تھوڑا سا اوپر کر تا ہے -

```
document.write('The is a');
var mystr = "2";
document.write(mystr.sup());
```

Next | Previous

sub Function

Sub Function کریکٹر کو لائین سے نیچے کی جانب ظاہر کرتا ہے

-

```
document.write('H');
var mystr = "2";
document.write(mystr.sub());
document.write('O');
```

H_2O

substring Function

یہ فنکشن ویلیو سے چند کریکٹر نکالنے کے کام آتا ہے دی گئی مثال میں شروع کے 3 کریکٹر کے بعد سے تمام کریکٹر آپ کو نظر آینگے

-

```
var mystr="JavaScript";
document.write(mystr.substring(3));
```

Next | Previous

valueOf Function

Valueof Function آپ کو Variable کی ویلیو ریٹرن کرے گا -

```
var mystr="JavaScript";
document.write(mystr.valueOf());
```

اب تک آپ نے String Variable کے ساتھ جتنے بھی فنکشن پڑھیں ہیں وہ سب String Function تھے -

Math Object

JavaScript میں حسابی عمل بھی کر سکتے ہیں جیسے دو نمبروں میں سے کون سی ویلیو بڑی ہے یا کون سی ویلیو چھوٹی ہے اور اس کے علاوہ PI, Log, Sin, Cos اور اس کے علاوہ مختلف قسم کے حسابی عمل بھی کر سکتے ہیں دی گئی مثال میں آپ کو PI پائی کی ویلیو ریٹرن ہو گی -

```
document.write(Math.PI);
```

Next | Previous

Absolute Value

منفی ویلیو کو مثبت دکھانے کے لیئے abs function کو استعمال کرتے ہیں مثال میں منفی ویلیو 5.72- ہے لیکن جب یہ براوزر پر دیکھیں گے تو مثبت نظر آئے گی -

```
document.write(Math.abs(-5.72));
```

Ceil Function

Ceil Function ویلیو کو راونڈ کرنے کے لیئے استعمال ہوتا ہے جیسے 5.1 کو یہ 6 میں تبدیل کر دے گا Point کے بعد Zero کے علاوہ کچھ بھی ہو یہ راونڈ کر دیتا ہے -

```
document.write(Math.ceil(5.1));
```

max Function

اس فنکشن کی مدد سے آپ بڑی رقم کو معلوم کر سکتے ہیں -

```
var myval = 45;
var myval2 = 81;
document.write(Math.max(myval,myval2));
```

Next | Previous

min Function

اس فنکشن کی مدد سے آپ سب سے چھوٹی ویلیو کو حاصل کر سکتے ہیں -

```
var myval = 45;
var myval2 = 81;
document.write(Math.min(myval,myval2));
```

random Function

Random نمبر کو Generate کرنے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے یاد رہے کہ یہ فنکشن 0 سے 1 تک Decimal Value ریٹرن کرتا ہے نیچے دیا ہوا کوڈ لکھیں اور Press 5F کرتے جائیں نمبر تبدیل ہوتے جایئں گے -

```
document.write(Math.random());
```

round Function

یہ فنکشن ویلیو کو Decimal عشاریہ 5. کے بعد سے راونڈ کرنے کے لیئے استعمال کیا جاتا ہے اگر ویلیو 5. سے کم ہو گی تو ویلیو راونڈ نہیں ہو گی -

```
document.write(Math.round(5.5));
document.write(Math.round(5.4));
```

Next | Previous

pow Function

کسی بھی Number کو کتنی دفعہ اسی Number سے ضرب کیا جائے جیسے 2 کو 4 دفعہ ضرب کیا جائے اور جواب آجائے اس کے لیئے اس فنکشن کو استعمال کرتے ہیں -

```
2 x 2 x 2 x 2 = 16
```

اوپر دی گئی مثال میں 2 کو 4 دفعہ ضرب کر کے جواب لیا گیا ہے اور اسی کام کو Pow Function سے بھی کر سکتے ہیں -

```
document.write(Math.pow(2,4));
```

eval Function

جاوا اسکرپٹ میں اگر نمبر اسٹرنگ میں ہو اور آپ اس کو نمبر ویلیو میں تبدیل کرنا چاہتے ہیں اس کے لیئے Eval Function استعمال کیا جاتا ہے -

```
document.write(eval("2+5"));
```

Next | Previous

isNaN Function

یہ فنکشن آپ کو بتاتا ہے کہ یہ ویلیو نمبر ہے یا نہیں isNaN کا معنی ہے is Not a Number یعنی دی گئی ویلیو نمبر نہیں ہے نیچے دی گئی مثال میں دی گئی نمبر ہے اس لیئے یہ فنکشن آپ کو False Return کرے گا کیونکہ یہ ایک نمبر ویلیو ہے -

```
document.write(isNaN(2));
```

نیچے دی گئی مثال میں اسٹرنگ ویلیو دی گئی ہے اس لیئے یہ فنکشن آپ کو True Return کرے گا کیونکہ یہ نمبر نہیں ہے -

```
document.write(isNaN("a"));
```

Next | Previous

parseInt Function

یہ فنکشن ایک اسٹرنگ ویلیو کو نمبر میں تبدیل کرتا ہے مثال میں ویلیو اسٹرنگ سے نمبر میں تبدیل ہو گئی ہے آپ سوچ رہے ہونگے کہ eval function بھی یہی کام کرتا ہے لیکن دونوں میں فرق ہے eval function اکسپریشن Expression کو نمبر میں تبدیل کرتا ہے parseInt فنکشن ایک نمبر کو اسٹرنگ سے نمبر میں تبدیل کرتا ہے -

```
document.write(parseInt("5"));
```

parseFloat Function

اگر اسٹرنگ ویلیو پوائنٹ میں ہو آپ اس کو نمبر میں تبدیل کرنا چاہتے ہے parseFloat function کو استعمال کرتے ہوئے آپ اس کو با آسانی تبدیل کر سکتے ہیں دی گئی مثال کو براوزر پر دیکھیں -

```
document.write(parseFloat("5.45"));
```

Next | Previous

const Keyword

JavaScript میں Variable کی ویلیو کو Fix رکھنے کے لیئے const keyword کو استعمال کیا جاتا ہے مثال کی پہلی لائین میں Variable کے نام سے پہلے const لکھا گیا ہے اس سے ویلیو Fix ہوجائے گی اس کے بعد ویلیو کو تبدیل کیا گیا ہے لیکن براوزر پر JavaScript ہی پرنٹ ہوگا -

```
const myname = 'JavaScript';
myname = 'Programing';
document.write(myname);
```

Events in JavaScript

جاوا اسکرپٹ میں ایونٹس بھی ہوتے ہیں ایونٹ مواقعے کو کہتے ہیں جیسے بٹن پر کلک یا کرنا ٹیکسٹ بوکس پر ٹائپ کرنا یا پیج لوڈ ہونا یا کسی ایلیمنٹ پر ماوس لے جانا یا ماوس ہٹالینا اس کے علاوہ اور بہت سے ایونٹ ہیں - ایونٹ کو آپ ٹیگ کے اندر دیکھ سکتے ہیں - جیسے آپ چاہتے ہیں کہ جب آپ کا پیج لوڈ ہو براوزر پر میسج آجائے اس

کے لیئے آپ کو بوڈی ٹیگ کے اندر Onload Event کو استعمال کرنا ہوگا ۔

```
<body onload="">
```

Next | Previous

اب آپ ایک فنکشن بنایئں اور اس کو پیج لوڈ پر کال کریں مثال میں mymsg کے نام سے فنکشن بنایا گیا ہے اور اس کو Body Tag کے اندر Onload Event میں کال کیا گیا ہے ایک بات یاد رہے کہ ایونٹ ہمیشہ Fire یا Trigger ہوتے ہیں اور فنکشن ہمیشہ کال ہوتا ہے جیسے ہی آپ کا پیج براوزر پر لوڈ ہو گا تو اون لوڈ ایونٹ فائر یا ٹریگر ہو گا اور اس کے ساتھ ہی mymsg فنکشن کال ہوگا اس طرح سے آپ تمام فنکشن کو جاوا اسکرپٹ میں بنایئں گے اور اس کو کسی بھی ایونٹس کے ساتھ استعمال کریں گے یہ اس کی پہلی مثال تھی آگے اور بھی مثالیں ہیں ہر مثال کو بار بار دوہرایئں تاکہ آپ کو اچھی سے طرح سمجھ آجائے ایک بات یاد رہے کہ Events میں Function ہی کال ہو گا ۔

```
<script type="text/javascript">
function mymsg()
{
    alert('This message shown on page load.');
}
</script>

<body onload="mymsg();">
```

Next | Previous

Click Event

اس مثال میں Button پر کلک کر کے فنکشن کو کال کریں گے دیئے گئے کوڈ میں آپ دیکھ رہے ہیں کہ Onclick Event کے اندر ہی میسج بوکس یعنی جاوا اسکرپٹ کا alert function ڈائیریکٹ کال کیا گیا ہے -

```
<input type="button" value="Click Me"
 onclick="alert('Click event trigger.');" />
```

آپ اس طرح بھی جاوا اسکرپٹ لکه سکتے ہیں اس کو لکھنے کا دوسرا طریقہ نیچے دیا گیا ہے فنکشن کو کال کرنے کے لیئے پہلے فنکشن کا نام لکھتے ہیں اس کے بعد بریکٹ لگاتے ہیں پھر آخر میں Semicolon آتا ہے اور ان تمام کو Inverted Comma کے اندر لکھا جاتا ہے -

```
function mymsg()
{
   alert('Click event trigger.');
}

<input type="button" value="Click Me"
onclick="mymsg();" />
```

Next | Previous

Mouseover Event

جیسا کہ نام سے ظاہر ہے کسی ایلیمنٹ پر Mouse آجائے جب کچھ Action Perform ہو نیچے دی گئی مثال میں ایک ہیڈنگ دی گئی ہے جیسے ہی آپ کا ماوس اس ہیڈنگ پر آئے گا ہیڈنگ کا ٹیکسٹ تبدیل ہو جائے گا ۔

```
 function changeHeading()
 {
    document.getElementById('myheading').innerHTML =
'Change Heading';
 }

  <h1 id="myheading" onmouseover="changeHeading();">
  This is heading1</h1>
```

Mouseout Event

اوپر دی گئی مثال میں ٹیکسٹ کو تبدیل کیا گیا ہے لیکن اگر آپ یہ چاہتے ہیں کہ جب ماوس ہیڈنگ سے ہٹ جائے تو واپس پہلے والا ٹیکسٹ آجائے اس کے لیئے آپ کو اس ایونٹ کو استعمال کرنا ہو گا اس کی مثال آپ اگلے پیج پر آپ دیکھ سکتے ہیں ۔

Next | Previous

```
function changeHeading()
  {
    document.getElementById('myheading').innerHTML =
'Change Heading';
  }
  function firstHeading()
  {
    document.getElementById('myheading').innerHTML =
```

```
    'This is heading1';
  }

  <h1 id="myheading" onmouseover="changeHeading();"
  onmouseout="firstHeading();">This is heading1</h1>
```

Blur Event

TextBox پر ٹیکسٹ لکھنے کے بعد دوسرے TextBox پر کلک کرتے ہیں یا ٹیب پریس کرتے ہیں اس وقت یہ ایونٹ فائر ہوتا ہے اس کی مثال نیچے دی گئی ہے -

```
  <input type="text" onblur="window.alert('This is blur
event!');" />
  <br />
  <input type="text" /><br />
```

Next | Previous

Focus Event

اکثر آپ نے دیکھا ہو گا کہ جب آپ www.google.com پر جا کر TextBox پر کلک کرتے ہیں تو کرسر بلنک کرنا شروع کر دیتا ہے اس کو مطلب یہ ہے کہ آپ کا فوکس اس TextBox پر ہے جیسے ہی آپ TextBox پر کلک کرتے ہیں تو فوکس ایونٹ فائر ہوتا ہے آیئے اب اس کی مثال دیکھتے ہیں -

```
   Enter Your Name:
   <input type="text" onfocus="alert('Focus Event is
trigger');" />
```

Submit Event

یہ ایونٹ HTML Form کے Submit ہوتے وقت فائر ہوتا ہے -

```
<form onsubmit="alert('Submit Event Trigger');">
What is your name?<br />
<input type="text" /><br />
<input type="submit" value="Submit" />
</form>
```

Next | Previous

Unload Event

یہ ایونٹ پیج کے Unload یعنی Close ہونے کے بعد فائر ہوتا ہے -

```
    <body onunload="alert('Thank you for visit our
website.');">
```

Window Object

Window Object جاوا اسکرپٹ کا ایک اہم ٹوپک ہے اس Object کے اندر آپ کو بہت سے فنکشن اور پراپرٹی ملیں گے جن کی مدد سے آپ بہت سے کام کر سکتے ہیں - آپ نے اب تک alert function استعمال کیا ہے ہو سکتا ہے کہ آپ کو یہ بات نہیں پتا ہو گی کہ یہ فنکشن بھی Window Object کا ہی ہے -

```
window.alert('Welcome to window object');
```

alert function کے ساتھ آپ window لکھیں یا نہ لکھیں ایک ہی بات ہے alert function دونوں ہی صورتوں میں کام کرے گا -

Next | Previous

Window Location

کسی Website پر ڈائیریکٹ جانے کے لیئے اس Property کو استعمال کیا جاتا ہے -

```
  function website()
  {
     window.location = 'http://www.google.com';
  }
  <input type="button" value="Go to google"
onclick="website();" />
```

اس کے علاوہ آپ اس پراپرٹی کو دوسرے پیج پر جانے کے لیئے بھی استعمال کر سکتے ہیں -

```
   window.location = 'contact.htm';
```

Window Find

پیج پر ٹیکسٹ کو سرچ کرنے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے جیسے ہی آپ بٹن پر کلک کریں گے Find Window اوپن ہو جائے گی -

```
    <input type="button" value="Find"
onclick="window.find();" />
```

Next | Previous

Window Home

Browser کے Home Page پر جانے کے لیئے اس فنکشن کو استعمال کرتے ہیں -

```
<input type="button" value="Home Page"
onclick="window.home();" />
```

Window Print

Browser سے Web Page کا Printout لینے کے لیئے اس فنکشن کو استعمال کیا جاتا ہے جیسے ہی آپ بٹن پر کلک کریں گے آپ کے سامنے ایک Print Dialog آئے گا پھر آپ Ok Button پر کلک کریں گے تو Printer سے Print نکلنا شروع ہو جائے گا -

```
<input type="button" value="Print out"
onclick="window.print();" />
```

Window Open

JavaScript میں آپ ایک New Window بھی اوپن کر سکتے ہیں اس کے علاوہ نئی ونڈو کے ساتھ اور بھی بہت کچھ کر سکتے ہیں اس کا کوڈ اگلے پیج پر دیکھیں -

Next | Previous

نیچے دیئے ہوئے کوڈ میں New Window اوپن کی گئی ہے یاد رہے اوپن فنکشن کے اور بھی پیرامیٹر ہیں جس میں ونڈو کی Height, Width اور Scroll Bar شامل ہیں -

```
function openpage()
{
  window.open("new_page.htm");
}
```

```
    <input type="button" value="Open Page"
onclick="openpage();" />
```

نیچے دی ہوئی مثال میں جیسے ہی آپ بٹن پر کلک کریں گے تو آپ کا پیج ایک نئی ونڈو میں کھلے گا اور اس کی Height Width وہ ہو گی جو آپ نے دی ہے -

```
    function openpage()
    {

window.open("contact.htm","newWindow","width=500,height=3
00");
    }
    <input type="button" value="Open Page"
onclick="openpage();" />
```

Next | Previous

اس مثال میں آپ براوزر میں Menu Bar, Scroll Bar اور ToolBar لگا رہے ہیں ان تینوں کی ویلیو Yes ہے اگر No کریں گے تو تینوں براوزر پر Show نہیں ہونگے -

```
    function openpage()
    {

window.open("contact.htm","newWindow","width=500,height=3
00,
      menubar=yes,scrollbars=yes,toolbar=yes");
    }
    <input type="button" value="Open Page"
onclick="openpage();" />
```

نئی ونڈو کی آپ پوزیشن بھی بتا سکتے ہیں کہ Top Side سے کتنے Pixel چھوڑنا اور Left Side سے کتنے Pixel چھوڑنا ہے اس کے لیئے ScreenX اور ScreenY کو استعمال کیا جاتا ہے نیچے دیئے گئے کوڈ میں یہی بتایا گیا ہے

```
window.open("contact.htm","newWindow","width=500,height=300,

menubar=yes,scrollbars=yes,toolbar=yes,screenX=300,screenY=250");
```

Next | Previous

Window Close

JavaScript کو استعمال کرتے ہوئے آپ اپنے Browser کو Close اس فنکشن کی مدد سے کر سکتے ہیں جیسے ہی بٹن پر کلک کریں گے آپ کا براوزر بند ہو جائے گا -

```
  <form>
  <input type="button" value="Close"
onclick="window.close();" />
  </form>
```

Window MoveTo

براوزر کو Top اور Left Side سے Move کرنے کے لیئے MoveTo فنکشن کو استعمال کرتے ہیں Button پر Click کرتے ہی براوزر Move ہوگا -

```
<form>
  <input type="button" value="Move Page"
    onclick="window.moveTo(50,50);" />
</form>
```

یاد رہے کہ صرف ایک دفع ہی براوزر veMo ہو گا ۔

Next | Previous